

METHOD AND SYSTEM FOR ROSTERING USING MULTIPLE LAYERED AND SEGMENTED PHEROMONE-BASED COOPERATING AGENTS

Field of the Invention

[0001] The present invention generally relates to computing methods and systems for rostering, and more particularly to a system and method for rostering using multiple layered and segmented pheromone-based cooperating agents.

Background of the Invention

[0002] Rostering problems arise in any organization that needs round-the-clock coverage of tasks. Such organizations include health care facilities, service-desks of public utility companies, police and fire departments, transportation companies, shipping companies, and airline/airport operators. Scheduling a staff duty roster for covering the round-the-clock tasks is a highly constrained hard combinatorial optimization problem (COP). The rostering problem is further complicated by flexible work patterns, public holiday returns, annual and training and urgent leaves, high level of manpower balance, and even-distribution of duty shifts.

[0003] Rostering is different from scheduling. In scheduling, the duration of each task is given and resource and temporal constraints are also known. The main objective of scheduling is to find a start time for every task and, typically at the same time, to allocate proper resources for the task as well. Often, the main objective requires total costs to be minimized. In rostering, a roster generally defines a work sequence using a shift list on a rotating basis within a group based upon predetermined parameters. Also, rostering relates to shift distribution; this is more than just controlling time for performing tasks as required in scheduling. A roster seeks to balance staff within constraints imposed. In some instances, rostering may have to accommodate other hard and soft constraints due to individual preferences, capability and availability.

[0004] Rostering problem has been a lasting one. Traditionally, constraint programming, Tabu Search, Simulated Annealing, Column-Generation, and Knowledge-

Based Systems (KBS) have been used to solve rostering problems. These traditional techniques are impractical for complex rostering problems.

[0005] Recently, more advanced probabilistic search technologies such as Genetic Algorithms (GA) and Ant Colony Optimization (ACO, or ant systems) have been applied to rostering problems.

[0006] Genetic algorithms search for optimal solutions on the basis of the principles of natural evolution that depends substantially on fitness. In a GA system, solution of a problem is encoded in a chromosome. Each unit of the chromosome, known as a gene, is a solution variable with a defined range of values. Genetic algorithms operate on a population of encoded solutions or chromosomes in a selection-reproduction evolution cycle. The first generation of a selection-reproduction evolution cycle consists of a number of randomly generated chromosomes that are selected according to their fitness quality. The fit members selected from the first generation are put into a mating pool for reproduction from which a new generation is derived by combining (crossover) or altering (mutating) genes among individuals within the population. The selection-reproduction evolution cycle repeats until an optimal solution has been found or a number of predetermined evolution cycles have been performed.

[0007] Genetic algorithms are useful in solving many real world problems. However, a preliminary condition for successful application of genetic algorithms is that the chromosomes for encoding the solutions must be with a manageable length. In traditional genetic algorithms systems, a chromosome (C_k) represents a solution, and a gene (denoted as $g_{k,j}$, where $j=1$ to L_C , L_C is length of the chromosome) represents an element of the solution. In this modeling, the element can have X values, each value being statically represented by a value of the gene. There are numerous ways to arrange shifts in a roster period. Applying static gene modeling using genetic algorithms to roster these shifts may not be feasible when the total number of permutations is huge. Therefore, when chromosomes exceed a manageable length, it becomes unfeasible for traditional GA systems to provide optimal roster solutions.

[0008] Singapore patent 103338, invented by Zhu Chun Bao and assigned to Nanyang Polytechnic, describes a dynamic GA modeling method and GA-based rostering system. Unlike traditional GA system that uses genes to statically represent solution parameters, the new method considers genes as shifting factors and swapping factors. The

dynamic adjusting factors are used in GA evolution solution searching cycles by the GA system to adjust predefined shift matrix to generate a roster solution. However, due to lengthy iteration cycles of GA evolution, solution searching may not be efficient, particularly when the rostering problem is considerably large. In addition, a GA system sometimes has difficulty to find real optimal solutions because optimization level largely depends upon quality of individual solutions in the first generation, which are essentially randomly generated.

[0009] Another approach to solve combinatorial optimization problem (COP) has been developed using what is known as ant colony optimization (ACO). Briefly, the ant colony optimization (ACO) is a meta-heuristic that makes use of cooperating agents called artificial ants. The ants iteratively construct candidate solution to a combinatorial optimization problem. An individual ant constructs a candidate solution by starting with an empty solution, and then iteratively adding solution components, which is selected using local search heuristic and pheromone deposited, if any, until a complete candidate solution is generated. After the solution construction is completed, the ants give feedback on the solutions they have constructed by depositing pheromone on the searching paths using global pheromone updating rules. Solution components, which are better or partially better, are used by more ants, and therefore receive a higher amount of pheromone until an optimal solution is found.

[0010] The ACO meta-heuristic consists of three problem-solving steps. First, the optimization problem is defined into a virtual graph. Second, a local searching heuristic is applied to select the next node for an ant to move to until the ant completes its solution construction. Finally, when all ants have constructed a complete solution, global pheromone updating rules are applied to update pheromone on links of visited nodes.

[0011] Although the ACO meta-heuristic has been proven useful for solving different hard combinatorial optimization problems, it is not readily available for use to solve real-world combinatorial optimization problems as the problem definition, local searching heuristic and global pheromone updating rules are very much problem-dependent.

[0012] In addition, traditional ant colony optimization (ACO) technology utilizes simple cooperating agents (also named artificial ants) to search for solutions; and during the solution searching within a virtual graph, the communication medium used by

cooperating agents is the pheromone. Actually pheromone is trail density, in form of a decimal number, deposited on the searching paths by searching ants. Therein, the traditional pheromone contains very limited information. As a result, it is not able to solve complex sizeable rostering problems because one must take into account a large number of solution parameters, various constraints imposed and multiple optimization criteria in order to find an optimal roster solution.

[0013] Therefore, a need clearly exists for an improved ACO-based rostering system and method to search for optimal roster solutions with high efficiency in situations with large solution spaces, various constraints imposed and multiple optimization criteria that cannot be resolved by existing rostering systems.

Summary of the Invention

[0014] Therefore, one objective of the present invention is to provide an improved ACO-based rostering method and system for optimal roster solutions with high efficiency.

[0015] Another objective of the present invention is to provide an improved ACO-based rostering method and system that can be applied to the rostering problems with more than one optimization criteria.

[0016] In one aspect of the present invention, there is provided a method for rostering a plurality of staff with duty assignments within a roster period using multiple layered and segmented pheromone-based cooperating agents, wherein the duty assignment of each staff is represented by a shift list. In one embodiment, the method comprises the following operations of receiving rostering problem definition data; constructing a virtual graph that comprises a plurality of staff nodes, a plurality of organizational nodes organized in a layered structure, a plurality of shift balance groups and a plurality of shift lists, wherein each of the plurality of staff nodes is associated with one corresponding organizational node, one corresponding shift balance group, and one corresponding shift list; performing main roster scheduling cycles on the virtual graph by a plurality of the multiple layered and segmented pheromone-based cooperating agents, wherein the cooperating agents utilize shifting factors to alter the shift lists and determine a roster

solution; and performing fine-tuning operation on the roster solution from the main roster scheduling cycles by swapping factors, wherein a best solution is found.

[0017] In one embodiment of the rostering method, in the step of construction of the virtual graph, the virtual graph is constructed by the following operations: establishing multiple layered organizational hierarchy; establishing balance groups at each of the organizational hierarchy layer; establishing of the assignment of staff to a duty team and identifying each staff's balance group; establishing original shift lists; building various data objects and models including multiple-layered and segmented data structure to stores shift balance statuses and pheromone density information; pre-fixing shifts that can not be altered within the rostering period; constructing a matrix of base shift lists; and constructing a matrix of adjustable shift lists by excluding non-adjustable shifts from the matrix of base shift lists; thereby the virtual graph is constructed based on organizational hierarchy, multiple layered balance groups, staff assignment and the matrix of adjustable shift lists.

[0018] In another embodiment of the method, the step of performing main scheduling cycles comprises the following operations: creating and initializing a predetermined number of cooperating agents and all related memory storages; setting staff's scheduling flags based on shift balance and constraints satisfactions evaluated according to the intermediate shift lists against various balance groups and multiple layered organizational hierarchy; wherein the intermediate shift lists are constructed by the system initially, and thereafter by the searching agents during roster solution searching cycles; the scheduling flag is included in the staff nodes of the virtual graph; placing the cooperating agents on the first staff node of the virtual graph; wherein each of the cooperating agents carries out a roster solution searching cycle starting from the first staff node to the last staff node; constructing a matrix of roster solution shift lists according to the shifting factors determined by the cooperating agent; evaluating, according to constraints imposed and shift balance criteria specified, the constructed roster solution searched by the cooperating agent; wherein if the solution $\Omega^a_{\text{solution}}$ is better than the solution searched in this searching cycle $\Omega^{\text{NC}}_{\text{best}}$, then the searching module updates $\Omega^{\text{NC}}_{\text{best}}$ by assigning $\Omega^a_{\text{solution}}$ to $\Omega^{\text{NC}}_{\text{best}}$; incrementing the agent counter by one; wherein one searching cycle is finished when all the agents have completed their solution searching cycles; evaluating the roster solutions searched in cycle NC by comparing $\Omega^{\text{NC}}_{\text{best}}$ with the global best solution Ω_{best} searched so

far; wherein if $\Omega_{\text{best}}^{\text{NC}}$ is better than Ω_{best} , updates Ω_{best} with $\Omega_{\text{best}}^{\text{NC}}$; updating state of the shift balances in the virtual graph, and multiple-layered and segmented trail densities; thereby the scheduling flags for all staff nodes are updated according to the evaluation of the shift balances and constraints satisfaction at various layers and segments; incrementing the solution searching cycle counter by one; and stopping search if either the best roster solution has no improvement after N_{Bstop} cycles such as when $N_{\text{ni}} \geq N_{\text{Bstop}}$ or if the total number of roster scheduling cycles reaches N_{max} (when $\text{NC} \geq N_{\text{max}}$); whereby the best roster solution Ω_{best} searched so far is presented for the fine tuning operations.

[0019] In yet another embodiment of the method, the fine-tuning operation comprises the following operations: initializing the fine-tuning operation, where the best solution Ω_{best} , searched by either the main scheduling cycles or previously fine-tuning operations, is copied to Ω_{Fbest} ; creating and initializing a predetermined number of cooperating agents M_{F} , and all related memory storages; determining candidate swaps for every staff node and determining high & low swapping regions for all the staff nodes; setting the staff's scheduling flags based on swapping factors, the shift balance and constraints satisfactions evaluated according to the best roster solution received from previous scheduling or fine-tuning operation Ω_{base} ; placing cooperating agents on the first staff node to carry out a roster fine-tuning cycle starting from the first staff node to the last staff node; constructing a matrix of roster solution shift lists, $\Omega_{\text{solution}}^{\text{a}}$ according to the list of swaps determined by the agent; evaluating, according to constraints imposed and shift balance criteria specified, the roster solution $\Omega_{\text{solution}}^{\text{a}}$ searched by the agent; if the solution $\Omega_{\text{solution}}^{\text{a}}$ is better than the solution $\Omega_{\text{best}}^{\text{NC}}$ searched in this searching cycle, then updates $\Omega_{\text{best}}^{\text{NC}}$ by assigning $\Omega_{\text{solution}}^{\text{a}}$ to $\Omega_{\text{best}}^{\text{NC}}$; incrementing the agent counter by one; checking whether all the agents have completed their solution searching cycles so as to determine whether the current searching cycle indexed by NC is finished; evaluating the roster solutions fine-tuned in cycle NC; evaluating the global best solution Ω_{Fbest} searched in fine-tuning cycle NC; updating state of the shift balances in the virtual graph; calculating trails $\Delta[\Psi(\tau^{\text{W}})_{\text{k,g}}]$, $\Delta[\Psi(\tau^{\text{S}})_{\text{k,j,g}}]$, $\Delta[\Psi(\tau^{\text{T}})_{\text{k,j,i,g}}]$; updating multiple-layered and segmented trail densities $\Psi(\tau^{\text{W}})_{\text{k,g}}(t+\Delta t)$, $\Psi(\tau^{\text{S}})_{\text{k,j,g}}(t+\Delta t)$ and $\Psi(\tau^{\text{T}})_{\text{k,j,i,g}}(t+\Delta t)$ on the virtual graph; and determining whether the fine-tuning is finished; wherein for a "NO", another fine-tuning cycle is carried out; otherwise, if either the best roster solution has no improvement after N_{FBstop} cycles such as when $N_{\text{ni}} \geq N_{\text{FBstop}}$ or if the total number of roster scheduling cycles

reaches N_{Fmax} (when $NC \geq N_{Fmax}$), then the fine-tuning cycles is finished and the best roster solution Ω_{Fbest} searched so far is copied to Ω_{best} which is presented as a temporary output.

[0020] In another aspect, the present invention provides a rostering system that comprises a computer-readable medium for storing computer programs, parameters, intermediate data, and final output; and an algorithm embedded within the computer-readable medium for rostering a plurality of staff with duty assignments within a roster period using multiple layered and segmented pheromone-based cooperating agents, wherein the duty assignment of each staff is represented by a shift list.

[0021] One advantage of the invention is that the new method has enhanced and significantly improved the traditional ant colony optimization (ACO) technology by creating a new type of pheromone, e.g., multiple-layered and segmented pheromones; and a new type of cooperating agents, e.g., multiple-layered and segmented pheromone based cooperating agents. Therefore, it is possible to use the new ACO to search for optimal solution in complex and large solution space represented by the virtual graph.

[0022] The new type of pheromone contains rich information about manpower balance, constraint satisfaction and roster optimization level at various organization hierarchy layers, shift balance groups and per a time unit (day) during the rostering periods. Therefore, it can represent various solution parameters, optimization criteria and constraint information. The segmented pheromone is able to divide the solution path into predefined number of segments, which has advantages over a single unified pheromone, significantly improves local searching and global searching algorithms, increases solution searching efficiency, and maximizes satisfaction level of soft constraints and optimization criteria.

[0023] Computational experiments show that the new system is most efficient among other rostering systems. It is about 30 times faster than today's best rostering software available on the market and in AI and OR research communities. It can also provide most optimal solutions as well.

[0024] Another advantages of the invention is that the method of scheduling and fine-tuning staff duty roster using multiple-layered and segmented pheromone-based cooperating agents allows the rostering engine to be flexible and to automatically generate a solution based on a number of input parameters, and the solution is optimized based on the optimization criteria and multiple hard constrains and soft constraints imposed.

[0025] The objectives and advantages of the invention will become apparent from the following detailed description of preferred embodiments thereof in connection with the accompanying drawings.

Brief Description of the Drawings

[0026] Preferred embodiments according to the present invention will now be described with reference to the accompanying Figures, in which like reference numerals denote like elements.

[0027] FIG. 1 is an exemplary matrix representation of a roster having a plurality of shift lists for a roster period.

[0028] FIG. 2 is a functional block diagram of a rostering system using multiple-layered and segmented pheromone-based cooperating agents in accordance with one embodiment of the present invention.

[0029] FIG. 3 illustrates a two-phase approach to generate a roster in accordance with one embodiment of the present invention.

[0030] FIG. 4 illustrates dynamic shifting of a shift list based upon different shifting factors.

[0031] FIG. 5 illustrates dynamic shifting of a shift list with several partially fixed shifts.

[0032] FIG. 6 illustrates swapping of two individual shifts in a shift list of an individual based upon different swapping factors.

[0033] FIG. 7 shows an exemplary source graph constructed by the graph constructor of the rostering system of FIG. 2.

[0034] FIG. 8 illustrates the balance groups related to an exemplary station and sub-balance groups related to teams under the station.

[0035] FIG. 9 illustrates the minimum manpower requirement, the duty shift distribution and shift balance during the rostering period for an exemplary balance group at an exemplary balance layer.

[0036] FIG. 10 illustrates an exemplary solution-searching path, the multiple layered organizational hierarchies and their associated multiple-segmented pheromone densities deposited by the searching agents in the virtual graph.

[0037] FIG. 11 illustrates the cooperating agent object, staff node object, local searching heuristic and deposition of segmented and multiple-layered pheromone trail densities.

[0038] FIG. 12 is a general flowchart of the rostering method using multiple-layered and segmented pheromone based cooperating agents in accordance with one embodiment of the present invention.

[0039] FIG. 13 is a detailed flowchart of the main roster scheduling module of FIG. 12 in accordance with one embodiment of the present invention.

[0040] FIG. 14 illustrates the generic swapping method for fine-tuning the roster previously generated using shifting factors by the searching agents.

[0041] FIG. 15 illustrates the selective swapping method and introduces how the candidate swaps are used in the rostering system for fine-tuning the roster by the searching agents.

[0042] FIG. 16 is a detailed flowchart of a roster fine-tuning module of FIG. 12 in accordance with one embodiment of the present invention.

Detailed Description of the Invention

[0043] The present invention may be understood more readily by reference to the following detailed description of certain embodiments of the invention.

[0044] Throughout this application, where publications are referenced, the disclosures of these publications are hereby incorporated by reference, in their entireties, into this application in order to more fully describe the state of art to which this invention pertains.

[0045] In the following detailed description, specific details are set forth in order to provide a thorough understanding of the invention. However, it will be understood by those skilled in the relevant art that the present invention may be practiced without these

specific details. In other instances, well-known methods, procedures, components, and materials have not been described in detail so as not to obscure the present invention.

[0046] It is to be noted that the following description will use a hospital environment as an exemplary organization to illustrate the principles of the present invention. The hospital environment serves a good COP example herein because its roster has a structural hierarchy and complex constraints. For example, its employees work in different wards, different stations and different teams. In addition, its individuals have different skill levels and experience. Furthermore, external factors such as public holidays or sickness of themselves or their colleagues affect rostering of the various shifts. These different requirements and external factors are typical of most organizations. Hence, the present invention is not limited to the hospital environment as described herein.

[0047] Referring now to FIG. 1, there is shown an exemplary matrix representation of a roster **10** having a plurality of shift lists for a roster period of n days. A shift list **16** is indicated as a generic example of the plurality of shift lists. The roster period starts from a first day **12** to an n^{th} day **14**. Other than days, n can also be of a different time unit depending upon the roster period that is required. Each of the shift lists corresponds to a workload of an individual staff in the roster period. The total number of shift lists in the roster **10** corresponds to the total number of staff that is represented by 'm'. Each cell **18** in the roster **10** represents a shift that is performed by a staff on a particular day. The terms 'workload' and 'shift list' are clarified hereinafter.

[0048] Normally, working shifts in a roster are set on a rotating basis to provide round-the-clock coverage. There are usually two ways to rotate the working shifts, namely, two-rotating shifts (2RS) or three-rotating shifts (3RS). The 2RS consists of a morning shift (D) and an afternoon shift (E) whereas the 3RS consists of a morning shift (D), an afternoon shift (E) and a night shift (N). For a particular rostering problem, each of the shifts has a defined time period, although variations to the defined time period are possible. Here, D, E and N are working shifts, whereas off-days (OD), public holidays (PH) and annual leaves (AL) and training leaves (TL) are non-working shifts.

[0049] Most organizations that need round-the-clock coverage define shift patterns as base templates over a fixed period of time. For example, a weekly base template for the 2RS may be "D, D, D, E, E, HD, RD" where 'HD' refers to a half day and 'RD' refers to a rest day. An example of a 3RS with a bi-weekly template is "D, D, D, E, E, HD, RD, E, E,

N, N, N, HD, RD". The half day 'HD' can be further defined as morning half day shift (MH) and afternoon half day shift (AH) respectively. These exemplary base templates only define the number of each kind of shifts in the roster period, for example there are 3 Ds and 2 Es in 2RS; although the shifts contained in the template appear in some sort of sequence, the sequence is not important at this stage of workload specification.

[0050] Often, a standard base template such as a 2RS or a 3RS that is assigned to a staff has to be changed for different rosters because of leave taking, public holidays, and fixed schedules. Changes in manpower management also affect rostering too. In planning a new roster because of changes requested by a staff, a new shift list that is based on a base template must be constructed and assigned to that staff. In manual rostering, the shift lists are adjusted or re-arrange manually to ensure that staff and shifts are balanced daily during the whole roster period and at every level of organizational hierarchy.

[0051] Workload and staff assignment can be very complex and are not easily automated. This is because both technical factors such as workload balance, and social factors such as government legislation and various staffing constraints have to be taken into account. For example, government legislation may not allow rostering to discriminate against gender, race or disability. As for staffing constraints, the workload and staff assignment should preferably consider human factors such as staff experience and willingness to work in specific groups. Generally, computers have a problem with trying to convert such human factors into rules, especially when the decision-making process is more social or political than technical. Thus, it is desirable and acceptable to manually perform certain tasks such as defining workload, working hours, number of work shifts, organizing team members while leaving complex calculations for deriving an optimal roster to computer processors.

[0052] Referring now to FIG. 3, a roster is derived in two phases in accordance with one embodiment of the present invention. Generally, in the first phase or Phase I, workload, staff assignment and shift lists are determined. Workload refers to the amount of work to be done by a group of staff. In some instances, workload refers to the number of staff and total number of work shifts required in each team or at each location where round-the-clock coverage services are provided. To every individual, his/her workload is often represented in the form of base-template 62 in the roster period 60. The base template contains a list of standard working or non-working shifts. After the base

templates are determined, pre-processing (also named shift-setting) often needs to be carried out **64** to construct the shift list **66**. Such shift-setting includes inserting leaves **64b**, imposing fixed schedules **64a** and returning staff public holidays **64c**. Note that in the shift-list **66**, the pre-fixed shifts and leaves will not be altered by the rostering engine; the shifts of which the positions can be altered to achieve overall balance form an adjustable shift list **68**, e.g., the shifts **66e** to **66x** as shown in FIG. 3. After the shift list **66** has been determined, shift balancing is determined, and a duty roster **72** generated by the rostering engine **70** in the second phase or Phase II.

[0053] In one aspect, the present invention provides a rostering system based on an improved ant colony optimization (ACO) using multiple-layered and segmented pheromone based cooperating agents. FIG. 2 shows a functional block diagram of a rostering system **20** using the improved ant colony optimization (ACO) technology in accordance with one embodiment of the present invention. The rostering system **20** comprises a workload and shift setting optimizer and graph constructor **24**, an ACO-based rostering engine **26**, and at least one storage medium **30**. It is to be noted that the storage medium **30** is indicated separately as different parts **30a**, **30b**, **30c**, **30d**, **30e** according to data stored therein for illustrative purposes only. As shown in FIG. 2, the storage medium **30a** stores roster calendar information **22**; the storage medium **30b** stores staff information **32** and multi-layered team organizational information **34**; the storage medium **30c** stores base templates **36**, non-working shifts such as staff leaves and public holidays **38**, and fixed schedule information **40**; the storage medium **30d** stores constraint information **42**, ACO-based searching parameters **44** and weight factor settings **46**; and the storage medium **30e** stores intermediate pheromone densities and shift balance state **50** and agent searching log **52**.

[0054] Outputs of the rostering system **20** are rosters **28** as exemplified by the roster **10** of FIG. 1.

[0055] The workload and shift setting optimizer **24** receives the roster calendar information **22** from the storage medium **30a**, the staff information **32** and multi-layered team organizational information **34** from the storage medium **30b**, and the base templates **36**, non-working shifts such as staff leaves and public holidays **38**, and fixed schedule information **40** from the storage medium **30c**. The workload and shift setting optimizer **24** optimizes the workload and actual shift loading in terms of overall balance in work force

within the roster period. The workload and shift setting optimizer **24** passes workload and shift setting information in the form of a shift list matrix to the ACO-based rostering engine **26**.

[0056] Fixed schedule **40** refers to certain shift lists that have partially fixed shifts or that have all shifts fixed. Fixed shifts, partially or completely, are commonly required for various personal reasons such as when an individual has to work in a particular shift on a particular day. The rostering system **20** does not alter the fixed schedule **40**, but takes into account of the fixed working shifts when calculating overall shift balance.

[0057] The ACO-based rostering engine **26** is an ACO-based artificial intelligence (AI) program that receives constraint information **42**, ACO-based solution-searching and control parameters **44** and weight factor settings **46** from the storage medium **30d** to search for roster solution **28**. To achieve the optimal roster, the total number of available staff and working shifts should be evenly distributed daily within the roster period at every organizational hierarchical layer. The roster **28** generated by the rostering engine **26** can be sent for printing or integration with other database systems.

[0058] The constraint information **42** includes legal regulations, organizational rules and personal requirements for leaves. The constraint information **42** also includes personal wishes and staff pairing; most of which are categorized as soft constraints. For instance, some staff may wish to work on a preferred shift pattern, for example, D-RD-E pattern and weekend DE pattern. The D-RD-E constraint states that it is desirable to take a morning shift before rest day (RD) and an afternoon shift after the rest day. The weekend DE constraints states that if a person has to be on duty during a weekend day (Saturday and Sunday), it is better to arrange a D shift on Saturday and an E shift on Sunday. Another constraint example is that a senior staff has to supervise a junior staff. Hence, in this constraint example, both the senior staff and the junior staff must be paired in the same working shifts for the whole roster period.

[0059] The rostering system **20** arranges constraints, both hard and soft constraints, based on a constraint's precedence according to their weight factor settings **46**. Most of the hard constraints such as the number of work hours and the number of D, E and N shifts in the roster period would have been satisfied when an initial shift list is provided. The ACO rostering engine **26** then evaluates additional constraints that were not considered previously.

[0060] The ACO-based solution searching parameters **44** comprise the number of search agents and the maximum number of searching cycles. However, for a given rostering problem, these agent searching parameters **44** may be adjusted for better solutions by comparing searched results stored in the agent searching log **52** associated with the storage medium **30e**.

[0061] The rostering system **20** keeps the solution searching state, such as intermediate pheromone densities deposited in the virtual graph and shift balance state **50** during a solution searching cycle. The intermediate pheromone and shift balance information may be refreshed when a new search cycle starts.

[0062] As there are a large number of rostering factors to be considered in determining an optimal roster solution, and such rostering factors may not equally affect searching for the optimal roster solution, there is a need to set a weight factor for each of the rostering factors. The weight factor ranges from 0 (least important) to 1 (most important). The weight factor settings **46** are normally predetermined based on collection of domain knowledge but can be adjusted to suit specific rostering problems.

[0063] A roster calendar **22** is automatically created by a roster calendar generator based upon the roster period specified. The rostering system **20** assigns a roster identification (ID) that is unique to each roster period. The roster period is pre-defined because the length of a shift list **16** in FIG.1 depends on the roster period. The roster calendar information **22** includes a list of public holidays (PH), which is essentially important in handling PH returns.

[0064] The ACO-based rostering engine **26** is adapted to perform ACO-based solution searching cycles by cooperating agents. The solution searching comprises two steps: 1) main scheduling and 2) fine-tuning. At step of the main scheduling, the search agents adjust the initial shift list matrix that has one or more shift lists using shifting factors to thereby derive a roster **28**. The search agents use swapping factors to fine-tune the roster that previously generated in Step one for further optimizing the roster solution.

[0065] The dynamic adjusting factors adopted in the rostering system comprise a shifting factor and a swapping factor. FIG. 4 illustrates dynamic shifting of a shift list **16** for different values (X) of the shifting factor. An un-shifted original shift list **16** is shown as a reference **80a** for the dynamic shifting. In **80b**, the shifting factor has a value of zero

($X=0$) and no shifting occurs. In **80c**, **80d**, **80e**, **80f**, the shifting factor has values of one ($X=1$), two ($X=2$), $n-2$ and $n-1$, **82** respectively.

[0066] FIG. 5 illustrates dynamic shifting of a shift list **16** with fixed shifts. An unshifted shift list **16** shown as a reference **90a** has two fixed shifts **92c**, **92d** within the roster period. The dynamic shifting of the reference **90a** is shown sequentially in FIG. 5. An adjustable shift list **90b** is formed by removing the two fixed shifts **92c**, **92d** for storage. Thereafter, the dynamic shifting occurs for all adjustable shifts such as **92a**, **92b**, **92e**, . . . , **92f**, where the shift **90g** was positioned as the first element as shown in **90c** for a shifting factor **94** with a value of one ($X=1$). Upon completing the dynamic shifting, the two fixed shifts **92c**, **92d** marked with symbol $\otimes$, are inserted back into the shift list **16** as shown in **90d** in their respective original positions as in the reference **90a**. The resulting shift list **90d** after dynamic shifting process is the duty roster for a staff although it may be fine-tuned at later stage of the rostering process.

[0067] FIG. 6 illustrates swapping of two individual shifts in a shift list **16** based upon the swapping factor. An un-swapped shift list **16** is shown as a reference **110a**. In **110b**, two individual shifts at positions **112b**, **112e** are swapped within the roster period. These positions are indicated by a swapping factor pair (2, $n-1$). In **110c**, two individual shifts at positions **112a**, **112c**, indicated by a swapping factor pair (1, 3), are swapped within the roster period. In **110d**, two individual shifts at positions **112c**, **112g**, indicated by a swapping factor pair (3, n) are swapped within the roster period. Similar to the situation described in FIG. 3 and 5, it is noted that within a shift list, not all shifts can be swapped. Therefore it is necessary to construct an adjustable shift list on which the swapping factors can be applied and then after swapping, the removed shifts will be inserted into the adjustable shift list to generate the roster shift list.

[0068] Referring now to FIG. 7, a virtual source graph **200** for the exemplary hospital nurse rostering problem is created. Where the nursing staff **202d** (**216**, **218**, **220**), assigned to various work sites or locations are grouped in a multi-layered organizational hierarchy. For example, in the hospital environment, the organizational hierarchy may consist of three layers, e.g., ward layer **202a**, station layer **202b**, and team layer **202c**. A staff nurse is assigned to an i^{th} -team under a j^{th} -station in a k^{th} -ward, where the k^{th} -ward (W_k **204** in the graph) may have several stations (S^{Wk}_1 **206**, S^{Wk}_2 **208**, . . . , $S^{\text{Wk}}_{N_s}$ **210**), and each j^{th} -station ($j = 1$ to $N^{\text{Wk}}_{N_s}$) may have several teams, where $N^{\text{Wk}}_{N_s}$ is the number of

stations in Ward W_k . For example the station S^{Wk}_1 **206** has Teams T^{S1}_1 **212** ..., and Team T^{S1}_{Nt} **214**, where, N_t is the number of teams under Station S^{Wk}_{s1} **206**. As discussed above, a staff nurse $P_{k,j,i}$, who is assigned to the i^{th} team under the j^{th} station in a k^{th} ward, can be expressed as follows (where the symbol $\Rightarrow$ stands for “is-assigned -to”).

$$[0069] \quad P_{k,j,i} \Rightarrow (W_k, S^{Wk}_j, T^{Sj}_i) \quad (1)$$

[0070] In FIG. 7, person or staff nodes **216**, **218**, **220**, are denoted as P^{SjTp}_h , meaning that the person is assigned to the T_p -th team of the S_j station (by default, in the Ward W_k); $h = 1$ to N_p , where N_p is the number of persons in the T_p -th team.

[0071] FIG. 7 shows an exemplary source graph (G) **200** constructed by the graph constructor **24** of the rostering system **20** shown in FIG. 2. The source graph G **200** can be expressed as follows:

$$[0072] \quad G = (W, S, T, P, BG, SL) \quad (2)$$

[0073] Where W = a set of ward nodes **202a**; S = a set of station nodes **202b**; T = a set of team nodes **202c**; P = a set of staff nodes **202d**; BG = a set of staff balance groups (not shown in graph **200**); SL = a set of shift lists **202e**, each of the shift list being associated with each of the staff nodes.

[0074] As can be seen from FIG. 7, each staff **202d** is associated with a shift list **202e**. For example, staff P^{S1T1}_1 is associated with shift list **226**. Note that the shift lists may contain non-adjustable shifts, **222**, **224**, e.g. fixed shifts, or leaves applied which are marked with symbol $\otimes$ in the shift list **226**. Of course, some shift lists may not have any non-adjustable shift defined by the planner; for example, the shift list **228** associated with staff P^{S1T1}_2 **218** does not have any fixed shifts or leaves, which is fully adjustable.

[0075] Referring to FIG. 7, a staff is assigned to a position in the organizational hierarchy, e.g., a staff nurse (P^{S1T1}_1) **216** is assigned to a team (T^{S1}_1) **212** under a station (S^{Wk}_{s1}) **206** in the ward **204** (W_k). This staff assignment which is normally performed in the first phase or Phase I of the rostering process determines the number of staff required by each organization unit at each organizational layer. The Phase II adjusts matrix of the shift lists and determines the work shift balance within the roster period at every

organizational hierarchical layer. As the roster must provide manpower round-the-clock coverage, the number of the morning shifts (D), afternoon shifts (E) and night shifts (N) must reach a good balance.

[0076] Referring to FIG. 8, an organization unit, say Station S^{wk}_j **310** in Ward W_k ($k=1$ to N_w , where N_w is the total number of wards in a hospital) may have several shift balance groups SbG^{Sj}_g **308 (306a, 306b, 306c, 306d,..., 306x)** where $g = 1$ to N_{Sj} , and N_{Sj} is the number of shift balance groups in Station S^{wk}_j **310**. It is noted that staff are often categorized by grades SG_k , such as for example senior staff nurse (SSN), or staff nurse (SN), the index $k = 1$ to N_{SG} , where N_{SG} is the number of total staff grades defined in an organization according to their level of trainings and professional skills. The staff grades are further grouped into shift balance groups BG_g , ($g = 1$ to N_{BG}), where N_{BG} is the number of total shift balance groups. Each of the shift balance groups BG_g is a subset of the set of staff grades and consists of one or more staff grades. A shift balance group can be defined at ward, station or team level.

[0077] As shown in FIG. 8, for example, for Station S^{wk}_j **310**, there are a list of shift balance groups **306a, 306b, 306c, 306d, ..., 306x**. Each of the balance groups is generally denoted as SbG^{Sj}_g **306d**, where $g = 1$ to N_{Sj} . Each of the shift balance groups, SbG^{Sj}_g may contain one or more staff grades. For example, a station level shift balance group SbG^{Sj}_1 may contain staff grade SSN (senior staff nurse) and SN (staff nurse), expressed as $SbG^{Sj}_1 = (SSN, SN)$. Generally, a station level shift balance group SbG^{Sj}_g that contains one or more staff grades SG^{Sj}_k can be expressed in following Equation (3).

$$\mathbf{[0078]} \quad SbG^{Sj}_g = (SG^{Sj}_1, SG^{Sj}_2, \dots, SG^{Sj}_{N_{Sjg}}) \quad (3)$$

[0079] Where N_{Sjg} is the number of staff grades in the balance group SbG^{Sj}_g .

[0080] Similarly, the shift balance groups WbG^{wk}_g for Ward W_k and the balance groups TbG^{Sj}_g for Team T^{Sj}_i **312a, 312b, ..., 312x**, in the station S^{wk}_j **310** can also be constructed using the same method. For example, the Team T^{Sj}_1 **312a** of Station S^{wk}_j **310** consists of following shift balance groups, TbG^{Sjt1}_1 **316a**, TbG^{Sjt1}_2 **316a**, ..., TbG^{Sjt1}_{nt1} **316a**, where $nt1$ is the number of shift-balance groups in the Team T^{Sj}_1 . Generally, a team level shift balance group TbG^{Sj}_g **318** under Station S^{wk}_j **310** may consist of one or more staff grades SG^{ti}_i , which is expressed in following equation:

[0081] $TbG_g^{Sj} = (SG_1^{ti}, SG_2^{ti}, \dots, SG_{N_{tig}}^{ti})$ (4)

[0082] Where N_{tig} is the number of staff grades in the balance group TbG_g^{Sj} .

[0083] It should be noted that the set of staff balance groups at various organization hierarchy levels are determined by the workload and shift setting optimizer 24 and stored in the storage block 34 as shown in FIG. 2.

[0084] In order to provide round-the-clock coverage of duty shifts with required skills, a balance group, say SbG_1^{Sj} 306a, normally needs a minimum number of D, E and N shifts, 304a, e.g., the number of morning shifts, $D_{SbGSj-1}$ 302a, the number of afternoon shifts, $E_{SbGSj-1}$ 302b and number of night shifts, $N_{SbGSj-1}$ 302c. As can be seen from FIG. 8, Station balance group SbG_g^{Sj} 306d requires $D_{SbGSj-g}$ morning shifts, $E_{SbGSj-g}$ afternoon shifts and $N_{SbGSj-g}$ night shifts 304k; and SbG_{Nsj}^{Sj} 306x requires $D_{SbGSj-Nsj}$, $E_{SbGSj-Nsj}$ and $N_{SbGSj-Nsj}$ shifts 304x respectively. In general, for the jth Station of kth Ward, the numbers of D, E or N shifts required in the gth shift balance group, denoted as $(N_{D-Shifts}^S)_{k,j,g}$, $(N_{E-Shifts}^S)_{k,j,g}$ and $(N_{N-Shifts}^S)_{k,j,g}$ are expressed according to three identifiers: 1.) ward W_k , 2.) station S_j^{Wk} and 3.) station balance group SbG_g^{Sj} , which are further described in following equation.

[0085]
$$\left. \begin{aligned} (N_{D-Shifts}^S)_{k,j,g} &= f(W_k, S_j^{Wk}, SbG_g^{Sj}) \\ (N_{E-Shifts}^S)_{k,j,g} &= f(W_k, S_j^{Wk}, SbG_g^{Sj}) \\ (N_{N-Shifts}^S)_{k,j,g} &= f(W_k, S_j^{Wk}, SbG_g^{Sj}) \end{aligned} \right\} \quad (5)$$

[0086] Referring still to FIG. 8, a station may contain several teams, for example, Station S_j^{Wk} 310 has N_{Nt}^{Sj} teams 314, T_1^{Sj} 312a, T_2^{Sj} 312b, ..., T_{Nt}^{Sj} 312x. Similarly, for each team, such as the team T_1^{Sj} 312a, has a list of balance groups, such as TbG_1^{Sjt1} 316a, TbG_2^{Sjt1} 316b, ..., TbG_{nt1}^{Sjt1} 316x, where, nt1 is the number of teams in Team T_1^{Sj} 312. The numbers of D, E and N shifts 320 required in the team balance group TbG_1^{Sjt1} 316a are $D_{TbGt1-1}$, $E_{TbGt1-1}$ and $N_{TbGt1-1}$ 320. Generic representations of the numbers of D, E and N shifts required in the gth team balance group for the ith team of the jth station in the kth ward, denoted as $(N_{D-Shifts}^T)_{k,j,i,g}$, $(N_{E-Shifts}^T)_{k,j,i,g}$ and $(N_{N-Shifts}^T)_{k,j,i,g}$ are described as follows.

$$\begin{aligned}
 \text{[0087]} \quad & \left. \begin{aligned}
 (N_{D\text{-Shifts}}^T)_{k,j,i,g} &= f(W_k, S_j^{Wk}, T_i^{Sj}, TbG_g^{Sjti}) \\
 (N_{E\text{-Shifts}}^T)_{k,j,i,g} &= f(W_k, S_j^{Wk}, T_i^{Sj}, TbG_g^{Sjti}) \\
 (N_{N\text{-Shifts}}^T)_{k,j,i,g} &= f(W_k, S_j^{Wk}, T_i^{Sj}, TbG_g^{Sjti})
 \end{aligned} \right\} \quad (6)
 \end{aligned}$$

[0088] Shift balance groups can also be defined at ward level. A typical balance group at ward level is denoted WbG_g . The numbers of D, E and N shifts required in WbG_g , denoted as $(N_{D\text{-Shifts}}^W)_{k,g}$, $(N_{E\text{-Shifts}}^W)_{k,g}$ and $(N_{N\text{-Shifts}}^W)_{k,g}$ are expressed as follows.

$$\begin{aligned}
 \text{[0089]} \quad & \left. \begin{aligned}
 (N_{D\text{-Shifts}}^W)_{k,g} &= f(W_k, WbG_g) \\
 (N_{E\text{-Shifts}}^W)_{k,g} &= f(W_k, WbG_g) \\
 (N_{N\text{-Shifts}}^W)_{k,g} &= f(W_k, WbG_g)
 \end{aligned} \right\} \quad (7)
 \end{aligned}$$

[0090] Note that the ward balance groups are not shown in FIG. 8.

[0091] Referring now to FIG. 9, the number of required D shifts, e.g., $(N_{D\text{-Shifts}}^S)_{k,j,g}$, $= f_{DS}(W_k, S_j^{Wk}, SbG_g^{Sj})$ **356** defines the number of morning shifts required for balance group SbG_g^{Sj} for Station S_j^{Wk} in Ward W_k per day within the whole rostering period ($D_1, D_2, \dots, D_n$) **358**. However, initially the D shifts are often not evenly distributed. FIG. 9 illustrates such an example **350a**; where the shaded rectangles corresponding to D_1 to D_n are drawn to illustrate the accumulated D shifts per days. The actual D shift distribution function is denoted as $n_{D\text{-Shifts}}^S(D_d)_{k,j,g}$, where $d = 1, 2, 3, \dots, n$; n is total number of days in the roster period. Note that the index k, j and g are sometimes omitted for simplicity. As can be seen from FIG. 9, $n_{D\text{-Shifts}}^S(D_1)$ **352** is lower than $N_{D\text{-Shifts}}^S$; while $n_{D\text{-Shifts}}^S(D_3)$ **354** is higher than $N_{D\text{-Shifts}}^S$.

[0092] FIG. 9 **350b** shows the deviations, **362a, 362b, 362c, 362d, 362e**, between the expected minimum number of D shifts **366** and actual D shift distribution **364**. The standard deviation is used in the rostering system to evaluate the deviations between actual shift distributions and expected number of D shifts.

[0093] The number of D shifts on day D_d for the g th balance group in j th station of k th ward, is denoted as $[n_{D\text{-shifts}}^S(D_d)]_{k,j,g}$. There exists the deviation $[(\Delta s_D^S)_d]_{k,j,g}$ from each data point, i.e.,

$$\text{[0094]} \quad [(\Delta s_D^S)_d]_{k,j,g} = [n_{D\text{-shifts}}^S(D_d)]_{k,j,g} - [N_{D\text{-shifts}}^S]_{k,j,g} \quad (8)$$

[0095] It should be noted that the number of D shifts on day D_d for balance group SbG^{Sj}_g , denoted as $[n^S_{D-shifts}(D_d)]_{k,j,g}$, can be calculated as follows:

[0096]

$$[n^S_{D-shifts}(D_d)]_{k,j,g} = \sum_{p=1}^{N^{staff}_{k,j,g}} [U(D_d)^{shift}_{k,j,g}]^D_p \quad (9)$$

[0097] where the $N^{staff}_{k,j,g}$ is the number of staff who are grouped under the balance group SbG^{Sj}_g of Station S^{Wk}_j in Ward W_k . The shift variable $[U(D_d)^{shift}_{k,j,g}]^D_p$ is the shift contributed by staff p , who is in the g th balance group SbG^{Sj}_g , assigned to the j th station of k th ward on day D_d . The variable $[U(D_d)^{shift}_{k,j,g}]^D_p = 1$ if the value of the shift variable is a D shift; otherwise variable $[U(D_d)^{shift}_{k,j,g}]^D_p = 0$.

[0098] The equations (8) and (9) only compute the deviation $[(\Delta S^S_D)_d]_{k,j,g}$ on one day, e.g., Day D_d . The standard deviation $[(\sigma^S)_D]_{k,j,g}$ is used as a measure of the variation of shift distribution during whole roster period, where $[(\sigma^S)_D]_{k,j,g}$ is the measure of the variation of the D shift distribution of the i th balance group in j th station of the k th ward. The superscript symbol S stands for “station”, which can be changed to T for computing a team’s balance group’s standard deviation. Value of $[(\sigma^S)_D]_{k,j,g}$ is calculated as follows.

[0099]

$$[(\sigma^S)_D]_{k,j,g} = \left\{ \frac{1}{n} \sum_{d=1}^n [(\Delta S^S_D)_d]_{k,j,g} \right\}'_2 = \left\{ \frac{1}{n} \sum_{d=1}^n \left\{ [n^S_{D-shifts}(D_d)]_{k,j,g} - [N^S_{D-shifts}]_{k,j,g} \right\}^2 \right\}'_2 \quad (10)$$

[00100] It should be noted that the morning shift D distribution curve **364** in FIG. 9 **350b** and Equations (8), (9) and (10) only reflect the method of how to compute the D shift distribution derivations for the station balance group SbG^{Sj}_g for Station S^{Wk}_j in Ward W_k . For example in Equation (10), the subscript symbol D stands for “D shifts” which can be changed to E or N to compute standard deviations of the variation in E or N shift distributions. Similar equations can be deduced for computing deviations of shift distributions for team and ward balance groups. The equation for computing the standard

deviation of shift distribution at team level is briefly expressed as follows (indexes are same as the ones used in Equation (10)).

$$[\sigma^T]_D]_{k,j,i,g} = \left\{ \frac{1}{n} \sum_{d=1}^n [\Delta S_D^T]_d]_{k,j,i,g} \right\}^{\frac{1}{2}} \quad (11)$$

where the subscript symbol g is used to index the balance groups in ith team of jth station of the kth ward; other details are ignored in the description for simplicity purpose. Similar equation can be deduced to compute standard deviation of shift distribution at ward level, shown in Equation (12).

$$[\sigma^W]_D]_{k,g} = \left\{ \frac{1}{n} \sum_{d=1}^n [\Delta S_D^W]_d]_{k,g} \right\}^{\frac{1}{2}} \quad (12)$$

Referring to FIG. 10, **400a**, a roster solution searching by cooperating agents using shifting factors is illustrated. A multi-layered organizational hierarchy is drawn in a rectangle bordered with a dotted line **402**. The organizational hierarchy consists of a list of stations **404**, e.g., S^{Wk_1} **404a**, S^{Wk_2} **404b**, ..., and $S^{Wk_{Ns}}$ **404x**; each station further consisting of a number of teams **406**. For example, Station S^{Wk_1} **404a** consists of Team T^{S1_1} **406a**, ..., and $T^{S1_{Nt}}$ **406x**. FIG. 10 **400a** also shows that the duty staff **408** are assigned to respective teams **406**, each staff has an adjustable shift list **412** associated with. The adjustable shift list **412** (e.g., **412a**, **412b**, **412c**, ..., **412x**) is constructed by the Workload & Shift Setting Optimizer **24** (see FIG. 2) by removing the non-adjustable shifts, e.g., fixed shifts **222**, **224** shown in FIG. 7. The adjustable shift list for staff p is expressed as follows.

$$SL^p_{adj} = (S_{p,1}, S_{p,2}, S_{p,3}, \dots, S_{p,N_{adj}}) \quad (13)$$

where N_{adj} is the number of adjustable shifts in the adjustable shift list, SL^p_{adj} is less or equal to n (n is number of days in the roster period). It is noted that it is not necessary to adjust SL^p_{adj} if $N_{adj}=1$, or if $N_{adj}>1$, but the adjustable shift contains shifts which are of same type, e.g., for example, D shifts, or non-working shifts.

[00107] The solution searching starts at first staff node **408a**, and the search path **416** is illustrated in the FIG. 10. The shifting positions **410**, **414**, e.g., the values of the shifting factors, are indicated using black solid circles.

[00108] Referring to FIG. 10, **400b**, a cooperating agent utilized in the traditional ACO technology deposits a single unified pheromone density along the search path **416** it traveled at end of one search cycle; the single unified pheromone density is indicated with a dotted line **420**. This is not the case in rostering practice as mentioned in FIG.7, FIG. 8 and FIG.9. In the present invention, the pheromone densities deposited by searching agents which mainly reflect the level of shift balance, say D shifts, for various different balance groups are represented by a segmented curve **422**, each segment of the curve being drawn with different thickness **422a**, **422b**, **422x**, reflecting different levels of D shift balances $[n^S_{D\text{-shifts}(D_d)}]_{k,j,g}$. For example, the first segment **422a** starting from the staff node **424** to staff node **426** represents the mean D shift balance for Station S^{Wk}_1 of Ward W_k ; the second segment **422b** starting from node **428** to node **430** represents the shift balance for Station S^{Wk}_2 of Ward W_k ; and the last segment **422x** represents the shift balance for Station S^{Wk}_{Ns} . Note that there are various balance groups under each station as shown in FIG. 8. The segmented pheromone density curve **422** only refers to one balance group among many others (SbG^{Sj}_g , $g=1$ to N_{sj} and $j=1$ to N^{Wk}_s).

[00109] Referring to FIG. 11, the G_8 is a part of the virtual graph G **200** shown in FIG.7 and **400a** in FIG. 10 respectively. The partial graph G_8 consists of one station S^{Wk}_1 **450**, which further consists of several teams T^{S1}_1 **452**, ..., T^{S1}_{Nt} **454**. The standard deviation of the morning shift (e.g., D shifts) distribution for each balance group SbG^{Sj}_g ($g=1$ to N_{sj} , N_{sj} is the number of total balance groups for Station S^{Wk}_{sj} **480**) is denoted as $[(\sigma^S)_D]_{k,j,g}$, where station index $j=1$, balance group index $g=1$ to N_{sj} . As can be seen from FIG. 11, the standard deviations of D shift distribution after one searching cycle can be very much different among balance groups SbG^{Sj}_g , which are illustrated using solid lines **482a**, **482b**, . . . , **482x** with different thickness, where the thinner solid line **482b** represents lower value of the standard deviation, e.g., a better shift balance. The thicker lines **482a**, **482x** mean that the balance levels are relatively lower, e.g., standard deviations are higher.

[00110] Similarly, at team balance level, the standard deviations, $[(\sigma^T)_D]_{k,j,i,g}$ of the D shift distributions are different from one team, T^{Sj}_1 **472** to another one T^{Sj}_{Nt} **476**;

differences are also among balance groups indexed by g . For example, as can be seen from FIG 11, the standard deviations $[(\sigma^T)_D]_{k, 1, 1, 1}$ of D shift distributions for Team T^{Sj}_1 **472**, which are represented by solid line **474a**, $[(\sigma^T)_D]_{k, 1, 1, 2}$ **474b**, and $[(\sigma^T)_D]_{k, 1, 1, nt1}$ **474x** are different from each other. Similarly, the standard deviations $[(\sigma^T)_D]_{k, 1, 2, 1}$ of D shift distributions for Team T^{Sj}_{Nt} **476**, which are represented by solid line **478a**, $[(\sigma^T)_D]_{k, 1, 2, 2}$ **478b** and $[(\sigma^T)_D]_{k, 1, 2, nNt}$ **478x** are different. It should be noted that in FIG. 11, the station index is 1 and the ward index is k , which implies that the method can be applied to all wards in the hospital. The various levels of the shift balances are represented by multiple-layered pheromone densities in the invention shown in following formulae.

$$\begin{aligned}
 \text{[00111]} \quad & \left. \begin{aligned}
 (\tau^W)_{k, g} &= C^W_{k, g} / \{ (\sigma^W)_{k, g} + (\Delta C^W)_{k, g} \} \\
 (\tau^S)_{k, j, g} &= C^S_{k, j, g} / \{ (\sigma^S)_{k, j, g} + (\Delta C^S)_{k, j, g} \} \\
 (\tau^T)_{k, j, i, g} &= C^T_{k, j, i, g} / \{ (\sigma^S)_{k, j, i, g} + (\Delta C^T)_{k, j, i, g} \}
 \end{aligned} \right\} \quad (14)
 \end{aligned}$$

[00112] where $C^W_{k, g}$, $C^S_{k, j, g}$ and $C^W_{k, j, i, g}$ are weight factors for computation of pheromone densities for wards, stations and teams respectively; and $(\Delta C^W)_{k, g}$, $(\Delta C^S)_{k, j, g}$ and $(\Delta C^T)_{k, j, i, g}$ are small decimal numbers added to the equations to avoid “divided by zero” errors which may occur when the standard deviation is zero. As can be seen from Equation (14), the present invention utilizes a multiple-layered and segmented pheromone density data structure model. Comparing with traditional unified pheromone density model, Equation (14) stores much more information that is useful for search agents to carry out efficient roster solution searching.

[00113] FIG. 11 also illustrates the cooperating agent object **456** and staff node object **458**. The cooperating agent object contains following attributes: agent ID, agent name which is optional, list of staff nodes visited, and list of adjusting factors determined. The adjusting factor F^P_{adj} for staff p , is expressed as follows.

[00114]

$$F^P_{adj} = \begin{cases} X^P & (\text{shifting}) \\ Y^P(y_1, y_2) & (\text{swapping}) \end{cases} \quad (15)$$

[00115] The staff node object contains following information: Staff ID, (e.g., $staff_p$), staff Grade SG_p , assignment $P_{i, j, k}$, the staff's base shift pattern ID, adjustable shift list

SL_{adj}^p , and scheduling-flag F_s^p . A staff's scheduling flag F_s^p can be set to On or Off. If staff_p's scheduling flag is set to Off, the rostering engine **26** program will not alter this staff's shift list. For example, in FIG. 11, the agent **456a** checks the scheduling flag of the staff node **458b**, the agent **456b** will not try to choose any adjusting factors to alter the adjustable shift list associated with the staff node **458b** if its scheduling flag is set to Off, as shown in FIG. 11. The scheduling flag is set to Off for staff node **458d** too, therefore its shift list is will not be altered. On the other hand, for staff node p, if the scheduling flag F_s^p is on, the agent **456a** will choose an adjusting factor F_{adj}^p from the shift positions in the adjustable shift list associated with the staff_p to alter the shift arrangement in order to achieve overall shift balance and high level constraint satisfaction. As shown in FIG. 11, the scheduling flags for staff nodes **458a**, **458c**, and **458e** are On, therefore the shifting factors **460**, **462** and **464** are chosen respectively.

[00116] After a solution searching cycle being conducted by a cooperating agent, the rostering engine will evaluate the intermediate solution searched by the agent and determine whether the scheduling flag is to be set to "Off" or "On" for every individual staff node. The scheduling-flag F_s^p is determined according to the factors listed below.

[00117] a) The shift balances of the balance groups at team and station level, where the staff is assigned to. The shift balances are computed using equations 8 to 12

[00118] b) The evaluation of imposed constraints including horizontal constraints and vertical constraints detailed hereinafter.

[00119] For example, when the standard deviation $[(\sigma^S)_D]_{k,j,g}$ and $[(\sigma^S)_D]_{k,j,i,g}$ are equal to zero, and constraints are satisfied (where the Staff_p is assigned to the ith team of jth station of kth ward, and under gth balance group), the scheduling flag F_s^p is turned to "Off".

[00120] An optimal roster solution not only needs to achieve good balance in various shift balance groups at various layer of the organization hierarchy, but also needs to reach a reasonably good satisfaction of various constraints imposed, including hard and soft ones. These constraints can be categorized into two groups: 1) vertical constraints and 2) horizontal constraints.

[00121] The horizontal constraints refer to the constraints which are applied to the roster matrix 10 in FIG. 1 horizontally, meaning that these constraints are applied to the shift lists ($S_{k,1}, S_{k,2}, \dots, S_{k,d}, \dots, S_{k,n}$) **16**. Typical examples of such horizontal constraints are listed but not limited as follows:

[00122] a) If a staff takes an afternoon shift (E), he or she should not be taking a morning shift next day;

[00123] b) A sleep day shift (SD) should be arranged after three consecutive night shifts;

[00124] c) If the number of night shifts is more than one, these night shifts should be arranged consecutively.

[00125] The quantitative evaluation of the horizontal constraints for staff_p, denoted as $(CSh)^p$, can be calculated as follows:

[00126]

$$(CSh)^p = \sum_{i=1}^{N_{CSh}} f(CSh_i) \times Wh_i \quad (16)$$

[00127] where the evaluation value of $f(CSh_i)$ is 1 if the horizontal constraint CSh_i is satisfied, otherwise, $f(CSh_i)$ is 0; Wh_i is weight factor for constraint CSh_i ; the N_{CSh} is the number of horizontal constraints imposed on the rostering problem.

[00128] The vertical constraints refer to the constraints that are applied to the roster matrix 10 in FIG. 1 vertically, meaning that these constraints are applied to the staff ($Staff_1, Staff_2, \dots, Staff_k, \dots, Staff_m$). Typical examples of such vertical constraints are listed but not limited as follows:

[00129] a) Staff A and Staff B take same duty shift;

[00130] b) Staff C and Staff D take opposite duty shifts.

[00131] Similarly, the quantitative evaluation of the vertical constraints for Day_d, denoted as $(CSv)_d$, can be calculated as follows:

[00132]

$$(CSv)_d = \sum_{i=1}^{N_{CSv}} f(CSv_i) \times Wv_i \quad (17)$$

[00133] where the evaluation value of $f(\text{CSv}_i)$ is 1 if the vertical constraint CSv_i is satisfied, otherwise $f(\text{CSv}_i)$ is 0; Wv_i is weight factor for constraint CSv_i ; the N_{CSv} is the number of horizontal constraints imposed on the rostering timetabling problem. It should be noted that d in Equation (17) is a day index.

[00134] Considering the multiple-layered and segmented pheromone trail density components calculated using Equation (14) and constraint satisfaction discussed above, the summarized multiple-layered and segmented pheromone trail densities at time t are expressed as follows:

$$\begin{aligned}
 \Psi(\tau^W)_{k,g} &= (\tau^W)_{k,g} + (\text{CSv})^W_{k,g} + (\text{CSh})^W_{k,g} \\
 \Psi(\tau^S)_{k,j,g} &= (\tau^S)_{k,j,g} + (\text{CSv})^S_{k,j,g} + (\text{CSh})^S_{k,j,g} \\
 \Psi(\tau^T)_{k,j,i,g} &= (\tau^T)_{k,j,i,g} + (\text{CSv})^T_{k,j,i,g} + (\text{CSh})^T_{k,j,i,g}
 \end{aligned}
 \quad \left. \vphantom{\begin{aligned} \Psi(\tau^W)_{k,g} \\ \Psi(\tau^S)_{k,j,g} \\ \Psi(\tau^T)_{k,j,i,g} \end{aligned}} \right\} \quad (18)$$

[00136] It is noted that time notation t is omitted, and at start point of time ($t=0$), the pheromones are set to a very small decimal number (say 0.001), or set to zero. The superscripts used in Equation (18) means that the pheromone is computed at different levels: W is at ward level, S is at station level, and T is at team level. The subscripts g is the balance group index; k the ward index; j the station index; and i the team index. The constraint evaluation components indexed using k, j, i, g , refers to the constraint evaluation for the g th balance group in i th team of j th station of k th ward.

[00137] Referring still to FIG. 11, when a cooperating agent, Agent_a **456a** comes to the first staff node (Staff_p) **458a**, the agent must check whether the scheduling flag F_s^P of the Staff_p **458a** is turned On or Off. If the scheduling flag F_s^P is turned Off, the Agent_a **456a** will not try to alter the shift list that is associated with the staff. However, if the scheduling flag F_s^P is turned On, the Agent_a **456a** chooses a shifting factor X among the shift positions ($z_1, z_2, z_3, \dots, z_q, \dots, z_{N_{Pz}}$) in the adjustable list based on a probability function, where z_q is the shifting position, $q = 1$ to N_{Pz} , N_{Pz} is the number of shifting positions in the adjustable list that is associated with staff_p. To compute the probability for choosing a shifting factor, we introduce a weighted selection factor Q_z^P , which is computed as follows.

[00138]
$$Q_z^P = \Psi_z^P(\tau^W)_{k,g} \times W_k^W + \Psi_z^P(\tau^S)_{k,j,g} \times W_{k,j}^S + \Psi_z^P(\tau^T)_{k,j,i,g} \times W_{k,j,i}^T \quad (19)$$

[00139] where $\Psi_z^P(\tau^S)_{k,j,g}$ is the pheromone density computed using Equation (18) for staff_p with selection of shifting position, e.g., shifting factor $X = z$. Note that z is a variable, of which the value is from 1 to N_{Pz} . For a particular staff, say Staff_p, the indexes, k, j, i and g , for the ward, station, and team and respective balance groups are known because of the known assignment information about the Staff_p. W_k^W , $W_{k,j}^S$ and $W_{k,j,i}^T$ are weight factors at ward, station and team layers respectively, which can be determined by the users although the initial values of the weight factors were set and stored in the storage 44 already.

[00140] The probability for selecting a shifting factor from the adjustable shift list being associated with staff_p can be calculated using following equation:

[00141]
$$P_z^P = (Q_z^P) / (Q_{sum}^P) \quad (20)$$

[00142] where the value of probability, P_z^P can be ranged from 0 to 1. The value of Q_{sum}^P is computed as follows:

[00143]
$$Q_{sum}^P = \sum_{z=1}^{N_{Pz}} Q_z^P \quad (21)$$

[00144] Note that Equation (21) summarizes all the weighted selection factors Q_z^P which can be possibly chosen from the adjustable shift list.

[00145] When all cooperating agents have completed construction of their respective roster solutions in one roster solution searching cycle, the multiple-layered and segmented trail intensities are updated using the following:

[00146]
$$\left. \begin{aligned} \Psi(\tau^W)_{k,g}(t+\Delta t) &= \rho_{k,g} \times \Psi(\tau^W)_{k,g}(t) + \Delta[\Psi(\tau^W)_{k,g}] \\ \Psi(\tau^S)_{k,j,g}(t+\Delta t) &= \rho_{k,j,g} \times \Psi(\tau^S)_{k,j,g}(t) + \Delta[\Psi(\tau^S)_{k,j,g}] \\ \Psi(\tau^T)_{k,j,i,g}(t+\Delta t) &= \rho_{k,j,i,g} \times \Psi(\tau^T)_{k,j,i,g}(t) + \Delta[\Psi(\tau^T)_{k,j,i,g}] \end{aligned} \right\} \quad (22)$$

[00147] where $\rho_{k,g}$, $\rho_{k,j,g}$ and $\rho_{k,j,i,g}$ are vaporization factors; $\Delta[\Psi(\tau^w)_{k,g}]$, $\Delta[\Psi(\tau^s)_{k,j,g}]$ and $\Delta[\Psi(\tau^t)_{k,j,i,g}]$ are the added pheromone components. As an example, $\Delta[\Psi(\tau^w)_{k,g}]$ is calculated using following equation:

$$[00148] \quad \Delta[\Psi(\tau^w)_{k,g}] = \sum_{a=1}^M [\Psi(\tau^w)_{k,g}]_a \quad (23)$$

[00149] where $[\Psi(\tau^w)_{k,g}]_a$ is quantity of trail substance laid by search agent, Agent_a on the search path segment on the gth balance group in the wth ward (see Equation (18)). M is the total number of search agents, and a is agent index (a = 1 to M).

[00150] In another aspect of the present invention, there is provided a rostering method using multiple-layered and segmented pheromone based cooperating agents. FIG. 12 shows a main flowchart of the method 500 of roster solution searching using the rostering system 20 in accordance with one embodiment of the present invention. In particular, the method 500 is described using multiple-layered and segmented pheromone based cooperating agents.

[00151] In order to clearly explain the rostering method 500, the data declarations which are closely related to the rostering solution search process are briefly described as follows.

[00152] Ω_o = Matrix of original shift lists, which are retrieved from base templates;

[00153] Ω_{adj} = Matrix of adjustable shift lists constructed after shift setting;

[00154] Ω_{base} = Matrix of base shift lists as base for alteration using shifting factors and swapping factors;

[00155] Ω_{best} = Matrix of intermediate best shift lists (also called global best solution);

[00156] Ω_{best}^{NC} = Matrix of temporary best solution search in search cycle NC; and

[00157] $\Omega_{solution}$ = Matrix of solution shift lists.

[00158] The matrix of original shift lists, Ω_o is constructed using the data retrieved from the base templates 36 and base shift pattern ID stored in the staff object. The matrix

of base shift lists, Ω_{base} , contains the information of shift setting, including fixed shifts, PH returns, etc. The adjustable shift lists, Ω_{adj} , are constructed by removing non-adjustable shifts from the base shift list matrix Ω_{base} . During solution searching process, the cooperating agents determine the adjusting factors. Based upon the list of adjusting factors which are determined by M agents in the search cycle NC (where NC is a roster solution searching cycle counter), the rostering engine constructs M intermediate roster solutions. Among the M intermediate roster solutions, a temporary best solution search in the current search cycle NC, denoted as $\Omega_{\text{best}}^{\text{NC}}$ is found. $\Omega_{\text{best}}^{\text{NC}}$ is assigned to Ω_{best} if $\Omega_{\text{best}}^{\text{NC}}$ is better than Ω_{best} during solution searching process in any cycle. At end of the roster solution searching procedure, the overall best solution Ω_{best} is copied to Ω_{solution} as final output.

[00159] Referring still to FIG. 12, after starting **502**, in the initialization step **504**, the rostering system will read/receive rostering problem definition data including the staff information, manpower organizational information, staff work shift base template data, team assignment information. The system constructs the virtual graph, original shift list lists Ω_o and build various data objects and models, for example multiple-layered and segmented data structure to stores shift balance and pheromone density information. At the same time, the shift setting optimizer **24** is used for the planner to pre-fix night shifts, apply leaves and process PH returns. Thereafter, a matrix of base shift lists Ω_{base} and the matrix of adjustable shift lists Ω_{adj} is constructed.

[00160] After initialization **504**, the rostering system **20** then proceeds to perform main roster scheduling cycles in module **506** on the source graph by one or more cooperating agents. The search agents utilize shifting factors to alter the shift lists stored in Ω_{base} and then to derive a roster solution Ω_{best} . The main roster scheduling is conducted normally once only.

[00161] At step **508**, the fittest shift list solution resulting from the main rostering module **506** is stored in matrix Ω_{best} for roster fine-tuning operation. At same time, a number of roster fine-tuning searching parameters are created and initialized. For example, rostering system **20** defines and initializes a fine tuning counter NF, a maximum number of fine-tunings N_{NFmax} ; set NF to zero and N_{NFmax} to C_{NFmax} , where the number of fine-tunings, C_{NFmax} is predetermined by a user and can be any arbitrary number depending on requirements of the rostering system **20**. It is recommended that C_{NFmax} be an integer number ranged from 1 to 3, or up to 5 at most.

[00162] The method **500** continues at step **510** at which the intermediate best solution matrix Ω_{best} is copied to a base shift matrix Ω_{base} for fine-tuning. In performing agent based solution search cycling in the fine-tuning module at step **512**, individual shifts identified by the swapping factor are swapped for the selected shift lists. Specifically, the swapping factor provides a pair of swapping factors, each of the swapping factors is associated with a selected individual shift within each of the selected shift lists. It is to be noted that the rostering system **20** applies a selective swapping method which will be discussed in detail hereinafter. Unlike main scheduling, the fine-tuning may be performed several times (N_{NFmax} times) until an optimal roster solution is found or predefined condition is met.

[00163] After each cycle of the roster fine-tuning performed by M agents, the method **500** proceeds to step **514** to store the best solution, denoted as Ω_{best} , which is found in fine-tuning cycle NF . The system updates the fine-tuning cycle counter NF , e.g. NF is incremented by one.

[00164] In checking step **516** in FIG. 12, whether the roster fine-tuning will continue depends upon whether the roster fine-tuning cycling counter NF is equal or larger than NF_{max} . With a “NO” the roster solution searching method **500** returns to step **510**, where the newly determined best solution Ω_{best} is copied to Ω_{base} for another cycle of fine-tuning. Otherwise, with a “YES”, the roster solution searching method proceeds to next step **518** to output the roster solution Ω_{solution} (where $\Omega_{\text{solution}} = \Omega_{\text{best}}$) and then comes to the end **520** of the roster solution searching procedure.

[00165] Now there is provided a more detailed description in FIG. 13 of the roster main scheduling module **506** in the method **500** briefly discussed above. As shown in FIG. 13, the main scheduling module **506** starts at the step **602**, which is connected to the output of the step **504** in FIG. 12.

[00166] The module **506** proceeds to the initializing step **602** by creating and initializing the predetermined number of cooperating agents, namely allocating memory space for predetermined number of cooperating agent objects **456**, and emptying their lists of visited staff nodes IDs, initialize trail densities and the multiple-layered segmented data structures; and the module **506** also initializes all related memory storages at this step. The staff's scheduling flags are set based on the shift balance and constraints satisfactions evaluated according to the base shift list Ω_{base} . The scheduling flag is included in the staff

node. At this step, the main scheduling module **506** places M agents on the first staff node, $Staff_a$; several rostering parameters are assigned with values too, such as the solution searching cycle counter NC is set to 0; N_{max} is set o C_m and N_{Bstop} is set to an integer number range from 20 to 50, the agent counter a is set to zero; and memory for Ω_{best} is initialized.

[00167] Step **604** increments the agent counter a by one.

[00168] Again referring to FIG. 13, at step **606** the $Agent_a$ carries out a roster solution searching cycle starting from the first staff node to the last staff node. For every staff node, say $Staff_p$, $Agent_a$ checks if the staff node's scheduling flag is on, if the answer is a "YES", $Agent_a$ chooses a shifting factor among the adjustable shift list associated with the staff with a probability computed by using Equations (18-20); if answer is a "NO", $Agent_a$ simply skips the staff node. In any case, the roster scheduling module **506** moves $Agent_a$ to next staff node, $Staff_{p+1}$. It is noted that choosing the shifting factor is repeated in a loop for a while until the last staff node, $Staff_m$ is reached.

[00169] Upon choosing in step **606** the shifting factors for all staff, the roster scheduling module **506** constructs a matrix of roster solution shift lists, $\Omega^a_{solution}$ according to the shifting factors determined by $Agent_a$

[00170] At step **608**, the roster scheduling module **506** evaluates, according to constraints imposed and shift balance criteria specified, the roster solution $\Omega^a_{solution}$ searched by $Agent_a$. If the solution $\Omega^a_{solution}$ is better than the solution Ω^{NC}_{best} searched in this searching cycle (counter = NC), then the searching module **506** updates Ω^{NC}_{best} by assigning $\Omega^a_{solution}$ to Ω^{NC}_{best} .

[00171] It is noted that at end of the searching cycle of the first searching agent, e.g., the searching $Agent^1$, Ω^{NC}_{best} and Ω_{best} are both initialized by using Ω^{NC}_{best} (where $NC=1$), although Ω^{NC}_{best} and Ω_{best} may be updated later on due to that a better solution might be found by searching agents in subsequent searching cycles.

[00172] In checking step **610** in FIG. 13 whether all the agents have completed their solution searching cycles determines whether the current searching cycle indexed by NC is finished. With a "NO" the roster scheduling module **506** returns to step **604** to let next searching agent carry out its searching cycle. Otherwise, with a "YES", the roster scheduling module **506** proceeds to next step **612**.

[00173] In step **612**, the roster scheduling module **506** evaluates the roster solutions searched in cycle NC. It compares $\Omega_{\text{best}}^{\text{NC}}$ with the global best solution Ω_{best} searched so far and updates Ω_{best} with $\Omega_{\text{best}}^{\text{NC}}$ if $\Omega_{\text{best}}^{\text{NC}}$ is better than Ω_{best} . In this situation, a counter (N_{ni}) for roster solution searching cycles during which there is no roster solution improvement is initialized by setting $N_{\text{ni}} = 0$. Otherwise, the counter N_{ni} is incremented by one to denote that one roster solution searching cycle has been performed with no solution improvement.

[00174] Referring again to FIG. 13, and after step **612**, the roster scheduling module **506** proceeds to step **614** to update state of the shift balances in the virtual graph using Equation (8-12). The module **506** then calculates trails $\Delta[\Psi(\tau^{\text{W}})_{k,g}]$, $\Delta[\Psi(\tau^{\text{S}})_{k,j,g}]$, $\Delta[\Psi(\tau^{\text{T}})_{k,j,i,g}]$ by using Equation (23) and update multiple-layered and segmented trail densities $\Psi(\tau^{\text{W}})_{k,g}(t+\Delta t)$, $\Psi(\tau^{\text{S}})_{k,j,g}(t+\Delta t)$ and $\Psi(\tau^{\text{T}})_{k,j,i,g}(t+\Delta t)$ on the source graph using Equation (22). At this step, the scheduling flags for all staff nodes are updated too according to the evaluation of the shift balances and constraints satisfaction at various layers and segments. Further the roster scheduling module **506** sets time $t = t + \Delta t$, and searching cycle counter $\text{NC} = \text{NC} + 1$.

[00175] The roster scheduling module **506** continues to determine in step **616** whether searching cycle is finished. For a “NO”, the roster scheduling module **506** will return to step **602** to carry on another searching cycle. Otherwise, if either the best roster solution has no improvement after N_{Bstop} cycles such as when $N_{\text{ni}} \geq N_{\text{Bstop}}$ or if the total number of roster scheduling cycles reaches N_{max} (when $\text{NC} \geq N_{\text{max}}$), then the roster scheduling module **506** stops, and the best roster solution Ω_{best} searched so far is presented at step **618** in FIG. 13. Note that Ω_{best} is forwarded to step **510** in FIG. 12, where Ω_{best} is copied to Ω_{base} for further fine-tuning operations.

[00176] It is noted that the roster scheduling module **506** in FIG. 12 performs roster main scheduling. FIG. 13 illustrates detailed description of the main roster scheduling module **506**. As shown in FIG. 12, after roster main scheduling is competed, the rostering system **20** further performs one or more rounds of fine-tunings on the roster solution which is generated by the roster scheduling module **506**. Rather than random swapping, an intelligent selective swapping method is created in the present invention in order to achieve high level of shift balance and constraint satisfaction.

[00177] Referring to FIG. 14, an exemplary D shift distribution chart, $[n_{D\text{-shifts}}^S(D_d)]_{k,j,g}$, is shown, where data used to compute the shift distribution is retrieved from the base roster shift list matrix Ω_{base} that is generated by the roster scheduling method **500** at step **510**. Note that the D shift distribution chart, $[n_{D\text{-shifts}}^S(D_d)]_{k,j,g}$ illustrates the morning shift distribution for gth balance group in jth station of the kth ward. As can be seen from the D shift distribution chart, during the roster period **702** (D_1 to D_7), there are some deviations in the D shift distribution. As can be seen from FIG. 14, at D_3 , D_4 and D_6 , the accumulated numbers of D shifts **706**, $[n_{D\text{-shifts}}^S(D_3)]_{k,j,g}$ **706c**, $[n_{D\text{-shifts}}^S(D_4)]_{k,j,g}$ **706d** and $[n_{D\text{-shifts}}^S(D_6)]_{k,j,g}$ **706f** are equal to the required minimum number of D shifts ($N_{D\text{-shifts}}^S$) $_{k,j,g}$ which is indicated by a horizontal line **704**; however, $[n_{D\text{-shifts}}^S(D_1)]_{k,j,g}$ **706a** and $[n_{D\text{-shifts}}^S(D_2)]_{k,j,g}$ **706b** are higher than ($N_{D\text{-shifts}}^S$) $_{k,j,g}$; but $[n_{D\text{-shifts}}^S(D_5)]_{k,j,g}$ **706e** and $[n_{D\text{-shifts}}^S(D_7)]_{k,j,g}$ **706g** are lower than ($N_{D\text{-shifts}}^S$) $_{k,j,g}$. In FIG. 14, the un-balanced $[n_{D\text{-shifts}}^S(D_d)]_{k,j,g}$, either higher or lower than the required number of shifts ($N_{D\text{-shifts}}^S$) $_{k,j,g}$ are shaded in order to differentiate them from the balanced ones.

[00178] Often, it is very difficult to achieve high level of shift balance by simply using shifting factors which is described previously in the present invention. Therefore it is necessary to apply swapping factors to reach a high level of shift balance and constraint satisfaction. For example as shown in FIG. 14, Staff_{P1} has a shift list that contains following shifts: D, D, E, AL, RD, D, E **712** and Staff_{P2} has a shift list of D, D, D, D, PH, D, RD **714**. Note that RD (rest day), PH (public holiday returned), and AL (annual leave) are non-working shifts, and Staff_{P1} and Staff_{P2} are in the same shift balance group indexed by k, j, and g. There are two swaps indicated in the FIG. 14: (1) Staff_{P1}: D **708** $\Leftrightarrow$ RD **710**; and (2) Staff_{P2}: D **716** $\Leftrightarrow$ RD **718**. After swapping operation, the new shift balance chart **720**, $[n_{D\text{-shifts}}^S(D_d)]_{k,j,g}$ is perfect.

[00179] More generically, the swaps described above are described using following Equation (24):

$$\text{Staff}_{P_z} : \text{SW}(\text{id}) = \{\text{Shift}(D_x) \Leftrightarrow \text{Shift}(D_y)\} \quad (24)$$

[00181] In Equation (24), the index $P_z = 1$ to m (m is the number of staff), $\text{SW}(\text{id})$ stands for the swap identified by id , where id is an integer type of identifier. The functions, $\text{Shift}(D_x)$ and $\text{Shift}(D_y)$, return the shifts positioned at (D_x) and (D_y) respectively, the

adjustable shift list, which the swapping factors are applied to, is associated with Staff_{P_z} . The symbol $\Leftrightarrow$ reads as “swaps with”, for example, $\{\text{Shift}(D_x) \Leftrightarrow \text{Shift}(D_y)\}$ meaning that the shift positioned at D_x swaps with the shift at (D_y) . It should be clear that x must NOT be equal to y . Therefore the swap (1) and swap (2) indicated in FIG. 14 can be expressed using Equation (24) as follows:

[00182] (1) $\text{Staff}_{P_1} : \text{SW}(1) = \{\text{Shift}(D_2) \Leftrightarrow \text{Shift}(D_5)\}$

[00183] (2) $\text{Staff}_{P_2} : \text{SW}(1) = \{\text{Shift}(D_1) \Leftrightarrow \text{Shift}(D_7)\}$

[00184] It should be noted that in the above expressions, the swap id is related to individual staff only. If there exists more than one swap for an individual staff, say Staff_{P_1} , the values of swap ids should be a series of consecutive running numbers starting from 1.

[00185] Referring now to FIG. 15 **800a**, there is a list of possible swaps within the shift list **802** associated with Staff_{P_1} . The shifts which can be candidate shifts being swapped are **D 802a**, **D 802b**, **RD 802c**, and **E 802d**. The possible swaps for Staff_P can be listed as follows (refer to the day index **808** in FIG. 15):

[00186] (1) $\text{Staff}_{P_1} : \text{SW}(1) = \{\text{Shift}(D_1) \Leftrightarrow \text{Shift}(D_2)\}$

[00187] (2) $\text{Staff}_{P_1} : \text{SW}(2) = \{\text{Shift}(D_1) \Leftrightarrow \text{Shift}(D_5)\}$

[00188] (3) $\text{Staff}_{P_1} : \text{SW}(3) = \{\text{Shift}(D_1) \Leftrightarrow \text{Shift}(D_7)\}$

[00189] (4) $\text{Staff}_{P_1} : \text{SW}(4) = \{\text{Shift}(D_2) \Leftrightarrow \text{Shift}(D_5)\}$

[00190] (5) $\text{Staff}_{P_1} : \text{SW}(5) = \{\text{Shift}(D_2) \Leftrightarrow \text{Shift}(D_7)\}$

[00191] (6) $\text{Staff}_{P_1} : \text{SW}(6) = \{\text{Shift}(D_5) \Leftrightarrow \text{Shift}(D_7)\}$

[00192] If a random swapping method is used, theoretically any swap in the above list can be picked up by the searching agent as swapping factor. However, it does not make any contribution to shift balance to swap two shifts which are of the same type, for example to swap the shift **D 802a** and shift **D 802b**. It even can make the shift balance level worse to add a D shift to the region in which the number of D shifts is already higher than the required number of D shifts, for example $[n^S_{D\text{-shifts}}(D_1)]_{k,j,g}$ **706a** and $[n^S_{D\text{-shifts}}(D_2)]_{k,j,g}$ **706b**.

[00193] Therefore it is desirable to divide the staff_p's possible swapping shifts into two regions: 1) the region with all the D_x at which the number of D shifts is higher than the required, denoted as Φ_{high}^P ; and 2) the region with all the D_y at which the number of D shifts is lower than the required, as described as follows.

$$\begin{aligned}
 [00194] \quad & \Phi_{high}^P = \{ D_{H1}, D_{H2}, \dots, D_{HNp} \} \\
 & \Phi_{low}^P = \{ D_{L1}, D_{L2}, \dots, D_{LNp} \}
 \end{aligned}
 \quad \left. \vphantom{\begin{aligned} \Phi_{high}^P \\ \Phi_{low}^P \end{aligned}} \right\} \quad (25)$$

[00195] In Equation (25), HNp is the number of possible swapping shifts in region Φ_{high}^P ; LNp is the number of possible swapping shifts in region Φ_{low}^P . The system can generate meaningful swapping towards improvement of shift balance if D_x and D_y are in different regions, and the shifts returned by $\text{Shift}(D_x)$ and $\text{Shift}(D_y)$ are of different types.

[00196] Applying the above selective swapping method, among the six possible swaps shown in FIG. 15 **800b**, swap (1) and swap (6) are removed. Finally the candidate swaps, generally denoted as $SW_{z,}^{P1}$, where $z = 1$ to N_{SWp1} ; N_{SWp1} is the number of candidate swaps for Staff_p , are listed as follows:

$$[00197] \quad SW_{1,}^{P1} = (2) \text{ Staff}_{P1}: SW(2) = \{\text{Shift}(D_1) \Leftrightarrow \text{Shift}(D_5)\}$$

$$[00198] \quad SW_{2,}^{P1} = (3) \text{ Staff}_{P1}: SW(3) = \{\text{Shift}(D_1) \Leftrightarrow \text{Shift}(D_7)\}$$

$$[00199] \quad SW_{3,}^{P1} = (4) \text{ Staff}_{P1}: SW(4) = \{\text{Shift}(D_2) \Leftrightarrow \text{Shift}(D_5)\}$$

$$[00200] \quad SW_{4,}^{P1} = (5) \text{ Staff}_{P1}: SW(5) = \{\text{Shift}(D_2) \Leftrightarrow \text{Shift}(D_7)\}$$

[00201] It should be noted that the Staff_p 's scheduling flag in fine-tuning stage is set to zero if the number of candidate swaps for Staff_p is zero.

[00202] Similar to the main scheduling process, in fine-tuning cycle, when a cooperating agent comes to a staff node (say Staff_p), the agent must check whether the scheduling flag F_s^P is "On" or "Off". If scheduling flag F_s^P is turned off, the agent will not swap any shifts in the shift list that is associated with the staff. However, if the flag F_s^P is turned on, the agent chooses a swap, SW_z^P , among the candidate swaps ($SW_1^P, SW_2^P, SW_3^P, \dots, SW_{N_{SWp}}^P$) based on a probability function. To compute the probability for choosing a swap, we introduce a weighted swap selection factor SQ_z^P , which is similar to Equation (19) as follows.

[00203]
$$SQ_z^P = \Psi_z^P(\tau^W)_{k,g} \times W_k^W + \Psi_z^P(\tau^S)_{k,j,g} \times W_{k,j}^S + \Psi_z^P(\tau^T)_{k,j,i,g} \times W_{k,j,i}^T \quad (26)$$

[00204] In Equation (26), $\Psi_z^P(\tau^S)_{k,j,g}$ is the pheromone density computed using Equation (18) for $staff_p$ with the shift alteration generated by applying the swap SW_z^P . Note that z is a variable, value of which is from 1 to N_{SWpz} , the ward, station, and team and respective balance groups indexed by k, j, i and g are known because of the known assignment information about the $Staff_p$. W_k^W , $W_{k,j}^S$ and $W_{k,j,i}^T$ are weight factors at ward, station and team layers respectively, which can be determined by the users although the initial values of the weight factors were set and stored in the storage 44 already.

[00205] The probability for selecting a swap from the candidate swaps being associated with $Satff_p$ can be calculated using following equation:

[00206]
$$SP_z^P = (SQ_z^P) / (SQ_{sum}^P) \quad (27)$$

[00207] where the value of SP_z^P is ranged from 0 to 1; the value of SQ_{sum}^P is computed as follows:

[00208]
$$SQ_{sum}^P = \sum_{z=1}^{N_{SWpz}} SQ_z^P \quad (28)$$

[00209] Note that Equation (28) summarizes all the weighted swaps SQ_z^P of the candidate swaps which are associated with $Staff_p$. The global trail density-updating formulae for fine-tuning are the same with the ones used in main roster scheduling.

[00210] Referring to FIG. 16, a more detailed description of roster fine-tuning module 512 is provided. The roster fine-tuning module 512 is different from the main scheduling module 506 although there are some similarities between them. The roster fine-tuning module 512 starts at the step 902, where the best solution Ω_{best} , previously searched by either the main scheduling module 506 or by fine-tuning module 512, is copied to Ω_{Fbest} for subsequent fine-tuning operations described below.

[00211] The module 512 proceeds to the initializing step 902 by creating and initializing the predetermined number of cooperating agents M_F , namely allocating

memory space for predetermined number of cooperating agent objects, and emptying their lists of visited staff nodes IDs, initialize trail densities and the multiple-layered and segmented data structures; and the module **512** also initializes all related memory storages at this step.

[00212] The module **512** proceeds to the determining candidate swaps for every staff node using Equation (24), and determining high & low swapping regions Φ_{high}^P & Φ_{low}^P (Equation (25)) for all staff nodes. Further, the staff's scheduling flags are set based on swapping analysis, the shift balance and constraints satisfactions evaluated according to the best roster solution received from previous scheduling or fine-tuning operation Ω_{base} (where $\Omega_{\text{base}} = \Omega_{\text{best}}$). Note that the scheduling flag is included in the staff node. At this step, the fine-tuning module **512** places M_F agents on the first staff node, Staff_a ; several rostering parameters are assigned with values, such as the solution searching cycle counter NC is set to 0; $N_{F\text{max}}$ is set to C_{FM} and $N_{FB\text{stop}}$ is set to an integer number range from 10 to 30 depending upon complexity of the rostering problems, the agent counter a is set to zero.

[00213] Step **904** increments the agent counter a by one.

[00214] Again referring to FIG. 16, at step **906** the Agent_a carries out a roster fine-tuning cycle starting from the first staff node to the last staff node. For every staff node, say Staff_P , Agent_a checks if the staff node's scheduling flag is on; if the answer is a "YES", Agent_a chooses a swap from the candidate swap list with a probability computed using Equations 27 & 28; if answer is a "NO", Agent_a simply skips the staff node. In any case, the roster fine-tuning module **512** moves Agent_a to next staff node, e.g., Staff_{P+1} . It is noted that choosing the swapping factor is repeated in a loop for a while until the last staff node, Staff_m is reached.

[00215] Upon choosing in step **906** the swapping factors for all eligible staff, the roster fine-tuning module **512** constructs a matrix of roster solution shift lists, $\Omega_{\text{solution}}^a$ according to the list of swaps determined by Agent_a

[00216] At step **908**, the roster fine-tuning module **512** evaluates, according to constraints imposed and shift balance criteria specified, the roster solution $\Omega_{\text{solution}}^a$ searched by Agent_a (note that $a = 1$ to M_F). If the solution $\Omega_{\text{solution}}^a$ is better than the solution $\Omega_{\text{best}}^{\text{NC}}$ searched in this fine-tuning cycle (counter = NC), then the roster fine-tuning module **512** updates $\Omega_{\text{best}}^{\text{NC}}$ by assigning $\Omega_{\text{solution}}^a$ to $\Omega_{\text{best}}^{\text{NC}}$.

[00217] It is noted that at end of the fine-tuning cycle carried out by the first searching agent, e.g., the fine-tuning Agent¹, $\Omega_{\text{best}}^{\text{NC}}$ and Ω_{Fbest} are both initialized by using $\Omega_{\text{best}}^{\text{NC}}$, although $\Omega_{\text{best}}^{\text{NC}}$ and Ω_{Fbest} may be updated later on due to that a better solution might be found by fine-tuning agents in subsequent fine-tuning cycles.

[00218] In checking step 910 in FIG. 16 whether all the agents have completed their fine-tuning cycles determines whether the current fine-tuning cycle indexed by NC is finished. With a “NO” the roster fine-tuning module 512 returns to step 904 to let next fine-tuning agent carry out its fine-tuning cycle. Otherwise, with a “YES”, the roster fine-tuning module 512 proceeds to next step 912.

[00219] In step 912, the roster fine-tuning module 512 evaluates the roster solutions fine-tuned in cycle NC. It compares $\Omega_{\text{best}}^{\text{NC}}$ with the global best solution Ω_{Fbest} fined-tuned so far and updates Ω_{Fbest} with $\Omega_{\text{best}}^{\text{NC}}$ if $\Omega_{\text{best}}^{\text{NC}}$ is better than Ω_{Fbest} . In this situation, a counter (N_{ni}) for roster fine-tuning cycles in which there is no roster solution improvement is initialized by setting $N_{\text{ni}} = 0$. Otherwise, the counter N_{ni} is incremented by one to denote that one roster fine-tuning cycle has been performed with no solution improvement.

[00220] Referring again to FIG. 16, after step 912, the roster fine-tuning module 512 proceeds to step 914, to evaluate the global best solution Ω_{Fbest} searched in fine-tuning cycle NC and update state of the shift balances in the virtual graph using Equation (8-12). The module 512 then calculates trails $\Delta[\Psi(\tau^{\text{W}})_{\text{k,g}}]$, $\Delta[\Psi(\tau^{\text{S}})_{\text{k,j,g}}]$, $\Delta[\Psi(\tau^{\text{T}})_{\text{k,j,i,g}}]$ using Equation (23) and update multiple-layered and segmented trail densities $\Psi(\tau^{\text{W}})_{\text{k,g}}(t+\Delta t)$, $\Psi(\tau^{\text{S}})_{\text{k,j,g}}(t+\Delta t)$ and $\Psi(\tau^{\text{T}})_{\text{k,j,i,g}}(t+\Delta t)$ on the source graph using Equation (22). At this step, the roster fine-tuning module 512 sets time $t = t + \Delta t$, and fine-tuning cycle counter $\text{NC} = \text{NC} + 1$.

[00221] The roster fine-tuning module 512 continues to determine in step 916 whether the fine-tuning is finished. For a “NO”, the roster fine-tuning module 512 will return to step 902 to carry on another round of fine-tuning. Otherwise, if either the best roster solution has no improvement after N_{FBstop} cycles, such as when $N_{\text{ni}} \geq N_{\text{FBstop}}$, or if the total number of roster fine-tuning cycles reaches N_{Fmax} (when $\text{NC} \geq N_{\text{Fmax}}$), then the roster fine-tuning module 512 stops fine-tuning and proceeds to step 918, where the best roster solution Ω_{Fbest} searched so far is copied to Ω_{best} which is presented as a temporary output. Note that the temporary best solution Ω_{best} is either forwarded to step 518 as final roster solution if stop condition is met, or to step 510 in FIG. 12 and copied to Ω_{base} for

another round of fine-tuning if the number of fine-tunings NF has not reached the pre-determined $NF_{\max}$.

[00222] The present invention therefore provides the rostering system **20**, the method **500** which includes an improved ACO-based roster scheduling module **506** and a roster fine-tuning module **512**. The rostering system and method are executed using the computer program to overcome, or at least alleviate the problems of the prior art in terms of solution searching efficiency and overall optimization quality.

[00223] While the present invention has been described with reference to particular embodiments, it will be understood that the embodiments are illustrative and that the invention scope is not so limited. Alternative embodiments of the present invention will become apparent to those having ordinary skill in the art to which the present invention pertains. Such alternate embodiments are considered to be encompassed within the spirit and scope of the present invention. Accordingly, the scope of the present invention is described by the appended claims and is supported by the foregoing description.

CLAIMS

What is claimed is:

1. A method for rostering a plurality of staff with duty assignments within a roster period using multiple layered and segmented pheromone-based cooperating agents, wherein the duty assignment of each staff is represented by a shift list, said method comprising the following operations of:

receiving rostering problem definition data;

constructing a virtual graph comprising a plurality of staff nodes, a plurality of manpower organizational units organized in a multiple layered structure, a plurality of shift balance groups and a plurality of shift lists; wherein each of said shift lists is associated with each of said staff nodes;

performing main roster scheduling cycles on the virtual graph by a plurality of the multiple layered and segmented pheromone-based cooperating agents, wherein the cooperating agents utilize shifting factors to alter the shift lists and determine a roster solution; and

performing fine-tuning operation on the roster solution from the main roster scheduling cycles by swapping factors, wherein a best solution is found.

2. The method of claim 1, wherein the multiple layered and segmented pheromone-based cooperating agents utilize multiple layered and segmented pheromone densities that are deposited by the search agents along the search paths on the virtual graph to perform the probability-based solution searching cycles.

3. The method of claim 2, wherein the multiple layered and segmented pheromone densities stored in a multiple-layered and segmented data structure, are computed typically as follows:

$$\begin{aligned}
 (\tau^W)_{k,g} &= C^W_{k,g} / \{ (\sigma^W)_{k,g} + (\Delta C^W)_{k,g} \} \\
 (\tau^S)_{k,j,g} &= C^S_{k,j,g} / \{ (\sigma^S)_{k,j,g} + (\Delta C^S)_{k,j,g} \} \\
 (\tau^T)_{k,j,i,g} &= C^T_{k,j,i,g} / \{ (\sigma^S)_{k,j,i,g} + (\Delta C^T)_{k,j,i,g} \}
 \end{aligned}$$

where k , j and i are indexes for respective manpower organization layers; g is the index for shift balance groups which are specified at each organization layer.

4. The method of claim 1, wherein the shift list comprises a plurality of shifts specified within the roster period, wherein each shift of the shift list is assigned to one specific staff node.
5. The method of claim 4, wherein each of the staff nodes is placed at a position within a multiple layered organizational hierarchy.
6. The method of claim 1, wherein the roster period is a time span within which the plurality of staff rotate their duties to provide a round-the-clock coverage of services.
7. The method of claim 1, wherein, in the step of receiving rostering problem definition data, the rostering problem definition data include the staff information, organizational information, staff work shift base template data, and team assignment information.
8. The method of claim 1, wherein, in the step of construction of the virtual graph, the virtual graph is constructed by the following operations:
 - establishing multiple layered organizational hierarchy;
 - establishing balance groups at each of the organizational hierarchy layer;
 - establishing of the assignment of staff to a duty team and identifying each staff's balance group;
 - establishing original shift lists;
 - building various data objects and models including multiple-layered and segmented data structure to stores shift balance statuses and pheromone density information;
 - pre-fixing shifts that can not be altered within the rostering period;
 - constructing a matrix of base shift lists; and
 - constructing a matrix of adjustable shift lists by excluding non-adjustable shifts from the matrix of base shift lists;

thereby the virtual graph is constructed based on organizational hierarchy, multiple layered balance groups, staff assignment and the matrix of adjustable shift lists.

9. The method of claim 8, wherein the non-adjustable shifts includes pre-fixed shifts, and leaves.

10. The method of claim 1, wherein the step of performing main scheduling cycles comprises the following operations:

creating and initializing a predetermined number of cooperating agents and all related memory storages;

setting staff's scheduling flags based on shift balance and constraints satisfactions evaluated according to the intermediate shift lists against various balance groups and multiple layered organizational hierarchy; wherein the intermediate shift lists are constructed by the system initially, and thereafter by the searching agents during roster solution searching cycles; the scheduling flag is included in the staff nodes of the virtual graph;

placing the cooperating agents on the first staff node of the virtual graph; wherein each of the cooperating agents carries out a roster solution searching cycle starting from the first staff node to the last staff node;

constructing a matrix of roster solution shift lists according to the shifting factors determined by the cooperating agent;

evaluating, according to constraints imposed and shift balance criteria specified, the constructed roster solution searched by the cooperating agent; wherein if the solution $\Omega^a_{\text{solution}}$ is better than the solution searched in this searching cycle $\Omega^{\text{NC}}_{\text{best}}$, then the searching module updates $\Omega^{\text{NC}}_{\text{best}}$ by assigning $\Omega^a_{\text{solution}}$ to $\Omega^{\text{NC}}_{\text{best}}$;

incrementing the agent counter by one; wherein one searching cycle is finished when all the agents have completed their solution searching cycles;

evaluating the roster solutions searched in cycle NC by comparing $\Omega^{\text{NC}}_{\text{best}}$ with the global best solution Ω_{best} searched so far; wherein if $\Omega^{\text{NC}}_{\text{best}}$ is better than Ω_{best} , updates Ω_{best} with $\Omega^{\text{NC}}_{\text{best}}$;

updating state of the shift balance in the virtual graph, and multiple-layered and segmented trail densities; thereby the scheduling flags for all staff nodes are updated

according to the evaluation of the shift balance and constraint satisfaction at various layers and segments;

incrementing the solution searching cycle counter by one; and

stopping search if either the best roster solution has no improvement after N_{Bstop} cycles such as when $N_{ni} \geq N_{Bstop}$ or if the total number of roster scheduling cycles reaches N_{max} (when $NC \geq N_{max}$); whereby the best roster solution Ω_{best} searched so far is presented for the fine tuning operations.

11. The method of claim 10, wherein the step of creating and initializing includes operations of allocating memory space for a predetermined number of cooperating agent objects, emptying their lists of visited staff nodes IDs, and initializing trail densities and the multi-layered and multi-segmented data structures.

12. The method of claim 10, wherein the step of placing the cooperating agents further comprises operations including assigning values to solution searching cycle counter and agent counter.

13. The method of claim 10, wherein during the solution search by the cooperating agents, each cooperating agent checks if the staff node's scheduling flag is on; if the answer is a "YES", the agent chooses a shifting factor among the adjustable shift list associated with the staff with a probability; if answer is a "NO", the agent simply skips the staff node; then the agent moves to next staff node of the virtual graph.

14. The method of claim 13, wherein the probability is computed by

$$P_z^P = (Q_z^P) / (Q_{sum}^P)$$

where the value of probability, P_z^P , can be ranged from 0 to 1.

15. The method of claim 10, wherein the main roster scheduling is preferably conducted once only.

16. The method of claim 1, wherein the number of fine-tunings is predetermined by a user.

17. The method of claim 1, wherein the number of fine-tunings is any arbitrary number depending on requirements of a rostering system.

18. The method of claim 1, wherein in the fine-tuning operation, individual shifts identified by the swapping factor are swapped for further fine-tuning the roster solution generated previously in main scheduling phase.

19. The method of claim 1, wherein the fine-tuning operation is preferably performed several times until an optimal roster solution is found or predefined condition is met.

20. The method of claim 1, wherein the fine-tuning operation comprises the following operations:

initializing the fine-tuning operation, where the best solution Ω_{best} , searched by either the main scheduling cycles or previously fine-tuning operations, is copied to Ω_{Fbest} ;

creating and initializing a predetermined number of cooperating agents M_F , and all related memory storages;

determining candidate swaps for every staff node and determining high & low swapping regions for all the staff nodes;

setting the staff's scheduling flags based on swapping factors, the shift balance and constraint satisfaction evaluated according to the best roster solution received from previous scheduling or fine-tuning operation on Ω_{base} ;

placing cooperating agents on the first staff node to carry out a roster fine-tuning cycle starting from the first staff node to the last staff node;

constructing a matrix of roster solution shift lists, $\Omega^{\text{a}}_{\text{solution}}$ according to the list of swaps determined by the agent;

evaluating, according to constraints imposed and shift balance criteria specified, the roster solution $\Omega^{\text{a}}_{\text{solution}}$ searched by the agent; if the solution $\Omega^{\text{a}}_{\text{solution}}$ is better than the solution $\Omega^{\text{NC}}_{\text{best}}$ searched in this searching cycle, then updates $\Omega^{\text{NC}}_{\text{best}}$ by assigning $\Omega^{\text{a}}_{\text{solution}}$ to $\Omega^{\text{NC}}_{\text{best}}$;

incrementing the agent counter by one;
 checking whether all the agents have completed their solution searching cycles so as to determine whether the current searching cycle indexed by NC is finished;
 evaluating the roster solutions fine-tuned in cycle NC;
 evaluating the global best solution Ω_{Fbest} searched in fine-tuning cycle NC;
 updating state of the shift balances in the virtual graph ;
 calculating trails $\Delta[\Psi(\tau^W)_{k,g}]$, $\Delta[\Psi(\tau^S)_{k,j,g}]$, $\Delta[\Psi(\tau^T)_{k,j,i,g}]$;
 updating multiple-layered and segmented trail densities $\Psi(\tau^W)_{k,g}(t+\Delta t)$, $\Psi(\tau^S)_{k,j,g}(t+\Delta t)$ and $\Psi(\tau^T)_{k,j,i,g}(t+\Delta t)$ on the virtual graph; and
 determining whether the fine-tuning is finished; wherein for a “NO”, another fine-tuning cycle is carried out; otherwise, if either the best roster solution has no improvement after N_{FBstop} cycles such as when $N_{\text{ni}} \geq N_{\text{FBstop}}$ or if the total number of roster scheduling cycles reaches N_{Fmax} (when $\text{NC} \geq N_{\text{Fmax}}$), then the fine-tuning cycles is finished and the best roster solution Ω_{Fbest} searched so far is copied to Ω_{best} which is presented as a temporary output.

21. The method of claim 20, wherein in the step of creating and initializing a predetermined number of cooperating agents M_F , and all related memory storages, it has operations including allocating memory space for predetermined number of cooperating agent objects, emptying their lists of visited staff nodes IDs, and initializing trail densities and the multiple-layered and -segmented data structures.

22. The method of claim 20, wherein during the search of staff nodes, the cooperating agent checks if the staff node's scheduling flag is on; if the answer is a “YES”, the agent chooses a swap from the candidate swap list with a probability; if answer is a “NO”, the agent simply skips the staff node.

23. The method of claim 20, wherein the evaluating of the roster solutions fine-tuned in cycle NC compares $\Omega_{\text{best}}^{\text{NC}}$ with the global best solution Ω_{Fbest} fined-tuned so far and updates Ω_{Fbest} with $\Omega_{\text{best}}^{\text{NC}}$ if $\Omega_{\text{best}}^{\text{NC}}$ is better than Ω_{Fbest} .

24. A rostering system comprising:

computer-readable medium for storing computer programs, parameters, intermediate data, and final output; and

an algorithm embedded within the computer-readable medium for rostering a plurality of staff with duty assignments within a roster period using multiple layered and segmented pheromone-based cooperating agents, wherein the duty assignment of each staff is represented by a shift list;

wherein the algorithm comprises:

a workload & shift setting optimizer and graph constructor module for receiving rostering problem definition data and constructing a virtual graph; and

an improved ACO-based rostering engine with a main scheduling module and a fine-tuning module; wherein the main scheduling module performs main roster scheduling cycles on the virtual graph by a plurality of the multiple layered and segmented pheromone based cooperating agents, wherein the cooperating agents utilize shifting factors to alter the shift lists and determine a roster solution, and the fine-tuning module performs fine-tuning operation on the best roster solution from the main roster scheduling cycles by swapping factors, wherein a best solution is found.

25. The rostering system of claim 24 wherein the multiple layered and segmented pheromone-based cooperating agents utilize multiple layered and segmented pheromone densities that are deposited by the search agents along the search paths on the virtual graph to perform the probability based solution searching cycles.

26. The rostering system of claim 24, wherein the multiple layered and segmented pheromone densities stored in a multiple-layered and segmented data structure, are computed typically as follows:

$$\begin{aligned}
 (\tau^W)_{k,g} &= C^W_{k,g} / \{ (\sigma^W)_{k,g} + (\Delta C^W)_{k,g} \} \\
 (\tau^S)_{k,j,g} &= C^S_{k,j,g} / \{ (\sigma^S)_{k,j,g} + (\Delta C^S)_{k,j,g} \} \\
 (\tau^T)_{k,j,i,g} &= C^T_{k,j,i,g} / \{ (\sigma^S)_{k,j,i,g} + (\Delta C^T)_{k,j,i,g} \}
 \end{aligned}$$

where k , j and i are indexes for respective manpower organization layers; g is the index for shift balance groups which are specified at each organization layer.

27. The rostering system of claim 24, wherein the shift list comprises a plurality of shifts specified within the roster period, wherein each shift of the shift list is assigned to one specific staff node.

28. The rostering system of claim 27, wherein each of the staff nodes is placed at a position within a multiple layered organizational hierarchy.

29. The rostering system of claim 24, wherein the roster period is a time span within which a plurality of staff rotate their duties to provide a round-the-clock coverage of services.

30. The rostering system of claim 24, wherein, in the step of receiving rostering problem definition data, the rostering problem definition data include the staff information, organizational information, staff work shift base template data, and team assignment information.

31. The rostering system of claim 24, wherein, in the step of construction of the virtual graph, the virtual graph is constructed by the following operations:

establishing multiple layered organizational hierarchy;

establishing balance groups at each of the organizational hierarchy layer;

establishing of the assignment of staff to a duty team and identifying each staff's

balance group;

establishing original shift lists;

building various data objects and models including multiple-layered and segmented data structure to stores shift balance statuses and pheromone density information;

pre-fixing shifts that can not be altered within the rostering period;

constructing a matrix of base shift lists; and

constructing a matrix of adjustable shift lists by excluding non-adjustable shifts from the matrix of base shift lists;

thereby the virtual graph is constructed based on organizational hierarchy, multiple layered balance groups, staff assignment and the matrix of adjustable shift lists..

32. The rostering system of claim 31, wherein the non-adjustable shifts includes pre-fixed shifts and leaves.

33. The rostering system of claim 24, wherein the step of performing main scheduling cycles comprises the following operations:

creating and initializing a predetermined number of cooperating agents and all related memory storages;

setting staff's scheduling flags based on shift balance and constraints satisfactions evaluated according to the intermediate shift lists against various balance groups and multiple layered organizational hierarchy; wherein the intermediate shift lists are constructed by the system initially, and thereafter by the searching agents during roster solution searching cycles; the scheduling flag is included in the staff nodes of the virtual graph;

placing the cooperating agents on the first staff node of the virtual graph; wherein each of the cooperating agents carries out a roster solution searching cycle starting from the first staff node to the last staff node;

constructing a matrix of roster solution shift lists according to the shifting factors determined by the cooperating agent;

evaluating, according to constraints imposed and shift balance criteria specified, the constructed roster solution searched by the cooperating agent; wherein if the solution $\Omega^a_{\text{solution}}$ is better than the solution searched in this searching cycle $\Omega^{\text{NC}}_{\text{best}}$, then the searching module updates $\Omega^{\text{NC}}_{\text{best}}$ by assigning $\Omega^a_{\text{solution}}$ to $\Omega^{\text{NC}}_{\text{best}}$;

incrementing the agent counter by one; wherein one searching cycle is finished when all the agents have completed their solution searching cycles;

evaluating the roster solutions searched in cycle NC by comparing $\Omega^{\text{NC}}_{\text{best}}$ with the global best solution Ω_{best} searched so far; wherein if $\Omega^{\text{NC}}_{\text{best}}$ is better than Ω_{best} , updates Ω_{best} with $\Omega^{\text{NC}}_{\text{best}}$;

updating state of the shift balances in the virtual graph, and multiple-layered and -segmented trail densities; thereby the scheduling flags for all staff nodes are updated

according to the evaluation of the shift balances and constraints satisfaction at various layers and segments;

incrementing the solution searching cycle counter by one; and

stopping search if either the best roster solution has no improvement after N_{Bstop} cycles such as when $N_{ni} \geq N_{Bstop}$ or if the total number of roster scheduling cycles reaches N_{max} (when $NC \geq N_{max}$); whereby the best roster solution Ω_{best} searched so far is presented for the fine tuning operations.

34. The rostering system of claim 33, wherein the step of creating and initializing includes operations of allocating memory space for a predetermined number of cooperating agent objects, emptying their lists of visited staff nodes IDs, and initializing trail densities and the multi-layered and multi-segmented data structures.

35. The rostering system of claim 33, wherein the step of placing the cooperating agents further comprises operations including assigning values to solution searching cycle counter and agent counter.

36. The rostering system of claim 33, wherein during the solution search by the cooperating agents, each cooperating agent checks if the staff node's scheduling flag is on; if the answer is a "YES", the agent chooses a shifting factor among the adjustable shift list associated with the staff with a probability; if answer is a "NO", the agent simply skips the staff node; then the agent moves to next staff node of the virtual graph.

37. The rostering system of claim 36, wherein the probability is computed by

$$P_z^P = (Q_z^P) / (Q_{sum}^P)$$

where the value of probability, P_z^P , can be ranged from 0 to 1.

38. The rostering system of claim 33, wherein the main roster scheduling is preferably conducted once only.

39. The rostering system of claim 24, wherein the number of fine-tunings is predetermined by a user.

40. The rostering system of claim 24, wherein the number of fine-tunings is any arbitrary number depending on requirements of a rostering system.

41. The rostering system of claim 24, wherein in the fine-tuning operation, individual shifts identified by the swapping factor are swapped for further fine-tuning the roster solution generated previously in main scheduling phase.

42. The rostering system of claim 24, wherein the fine-tuning operation is preferably performed several times until an optimal roster solution is found or predefined condition is met.

43. The rostering system of claim 24, wherein the fine-tuning operation comprises the following operations:

initializing the fine-tuning operation, where the best solution Ω_{best} , searched by either the main scheduling cycles or previously fine-tuning operations, is copied to Ω_{Fbest} ;

creating and initializing a predetermined number of cooperating agents M_F , and all related memory storages;

determining candidate swaps for every staff node and determining high & low swapping regions for all the staff nodes;

setting the staff's scheduling flags based on swapping factors, the shift balance and constraints satisfactions evaluated according to the best roster solution received from previous scheduling or fine-tuning operation on Ω_{base} ;

placing cooperating agents on the first staff node to carry out a roster fine-tuning cycle starting from the first staff node to the last staff node;

constructing a matrix of roster solution shift lists, $\Omega^a_{\text{solution}}$ according to the list of swaps determined by the agent;

evaluating, according to constraints imposed and shift balance criteria specified, the roster solution $\Omega^a_{\text{solution}}$ searched by the agent; if the solution $\Omega^a_{\text{solution}}$ is better than the

solution $\Omega_{\text{best}}^{\text{NC}}$ searched in this searching cycle, then updates $\Omega_{\text{best}}^{\text{NC}}$ by assigning $\Omega_{\text{solution}}^{\text{a}}$ to $\Omega_{\text{best}}^{\text{NC}}$;

incrementing the agent counter by one;

checking whether all the agents have completed their solution searching cycles so as to determine whether the current searching cycle indexed by NC is finished;

evaluating the roster solutions fine-tuned in cycle NC;

evaluating the global best solution Ω_{Fbest} searched in fine-tuning cycle NC;

updating state of the shift balances in the virtual graph ;

calculating trails $\Delta[\Psi(\tau^{\text{W}})_{\text{k,g}}]$, $\Delta[\Psi(\tau^{\text{S}})_{\text{k,j,g}}]$, $\Delta[\Psi(\tau^{\text{T}})_{\text{k,j,i,g}}]$;

updating multiple-layered and segmented trail densities $\Psi(\tau^{\text{W}})_{\text{k,g}}(t+\Delta t)$, $\Psi(\tau^{\text{S}})_{\text{k,j,g}}(t+\Delta t)$ and $\Psi(\tau^{\text{T}})_{\text{k,j,i,g}}(t+\Delta t)$ on the virtual graph; and

determining whether the fine-tuning is finished; wherein for a “NO”, another fine-tuning cycle is carried out; otherwise, if either the best roster solution has no improvement after N_{FBstop} cycles such as when $N_{\text{ni}} \geq N_{\text{FBstop}}$ or if the total number of roster scheduling cycles reaches N_{Fmax} (when $\text{NC} \geq N_{\text{Fmax}}$), then the fine-tuning cycles is finished and the best roster solution Ω_{Fbest} searched so far is copied to Ω_{best} which is presented as a temporary output.

44. The rostering system of claim 43, wherein in the step of creating and initializing a predetermined number of cooperating agents M_{F} , and all related memory storages, it has operations including allocating memory space for predetermined number of cooperating agent objects, emptying their lists of visited staff nodes IDs, and initializing trail densities and the multiple layered and segmented data structures.

45. The rostering system of claim 43, wherein during the search of staff nodes, the cooperating agent checks if the staff node's scheduling flag is on; if the answer is a “YES”, the agent chooses a swap from the candidate swap list with a probability; if answer is a “NO”, the agent simply skips the staff node.

46. The rostering system of claim 43, wherein the evaluating of the roster solutions fine-tuned in cycle NC compares $\Omega_{\text{best}}^{\text{NC}}$ with the global best solution Ω_{Fbest} fined-tuned so far and updates Ω_{Fbest} with $\Omega_{\text{best}}^{\text{NC}}$ if $\Omega_{\text{best}}^{\text{NC}}$ is better than Ω_{Fbest} .

47. The rostering system of claim 24, wherein the information stored in the computer-readable medium and used by the workload & shit setting optimizer and graph constructor module includes roster calendar information, staff information and multi layered team organizational information, and base templates, non-working shifts such as staff leaves and public holidays, and fixed schedule; and wherein the information stored in the computer-readable medium and used by the rostering engine includes constraint information, ACO-based evolution parameters and weight factor settings, and intermediate pheromone densities shift balance state and agent searching log.