

A SYSTEM AND A METHOD OF SCHEDULING

The present application claims priority to the Singapore patent application no. 10202108076X which is incorporated in its entirety by reference.

TECHNICAL FIELD

[0001] The present disclosure relates to resource allocation, and more particularly to a method of scheduling and a system thereof.

BACKGROUND

[0002] Scheduling problems, such as combinatorial optimization problems (COP), cover numerous real-world tasks in many fields such as communication, transportation, and manufacturing. It is often a challenge to solve COPs due to non-deterministic polynomial-time hardness (NP-hardness). When faced with a variety of constraints, the performance of conventional methods, including exact and heuristic algorithms, are generally limited by hand-crafted policies that are costly to design, and which require considerable trial-and-error and domain knowledge.

SUMMARY

[0003] In one aspect, the present disclosure provides a system for scheduling a plurality of airport ground handling resources, the system comprising: a scheduling model configured with a plurality of decision variables, the plurality of decision variables corresponding to at least one of the plurality of airport ground handling resources, each of the plurality of decision variables having a respective variable value; and a computational device. The computational device is configured to perform a method of: for a present iteration and based on a policy, selecting individual ones of the plurality of decision variables to form a subset, each of the plurality of decision variables forming the subset being configured to be updateable by a solver in a present iteration, each of the plurality of decision variables forming the subset being selected independently of another of the plurality of decision variables; determining a complement of the subset, wherein the complement and the subset are mutually exclusive, the complement being formed of other of the plurality of decision variables which are configured to be not updated by the solver in the present iteration;

providing the subset and the complement to the solver as inputs, the solver being configured to optimize the scheduling model to determine respective updated variable values of the subset in the present iteration, each of the respective variable values of the complement being held constant by the solver in the present iteration; and if a predetermined condition is not met, using the updated variable values of the subset and the variable values of the complement as the plurality of decision variables for a subsequent iteration, wherein a subsequent subset and a subsequent complement are determined from the plurality of decision variables in the subsequent iteration.

[0004] In one embodiment, the subset is non-mutually exclusive between the present iteration and the subsequent iteration. In one embodiment, a dimension of the subset is variable between the present iteration and the subsequent iteration. The method may further comprise, prior to optimizing the scheduling model, removing the respective variable values of the subset for the present iteration as a part of a destroy operation. In one embodiment, the subset is formed by inputting the respective variable values of the present iteration to a machine learning model and obtaining an output from the machine learning model.

[0005] The system may be configured such that the machine learning model is trained via an imitation learning-based method, wherein by use of a heuristic operator, a heuristic initial variable value is provided to the imitation learning-based method. The system may be configured such that the machine learning model is trained via a reinforcement-learning based method, and wherein a randomized initial value is provided to the reinforcement learning-based method. In one embodiment, the output is a probability distribution with a probability value corresponding to each of the decision variables of the present iteration, wherein each of the respective probability value corresponding to the subset is above a threshold value. The system may be configured to perform a plurality of the subsequent iteration until the predetermined condition is met.

[0006] In one embodiment, the plurality of airport ground handling resources is grouped into a plurality of functional groups, each of the plurality of functional groups being configured to perform at least one respective related task, and wherein the computational device is configured to schedule the plurality of functional groups to perform the at least one respective related task. In one embodiment, the computational device is configured to

schedule at least one of the plurality of functional groups to perform a plurality of related tasks, wherein the plurality of related tasks being a series of sequence related tasks. In one embodiment, the computational device is configured to schedule two or more of the plurality of functional groups to concurrently perform the at least one respective related tasks. One of the decision variables may correspond to at least one of: a movement of one of the plurality of airport ground handling resources from one aircraft to another aircraft; an availability of one of the airport ground handling resources; and a duration of service for the one aircraft for one of the plurality of airport ground handling resources.

[0007] According to another aspect, the present disclosure provides a method of scheduling resources, the method comprising: providing a scheduling model configured with at least one objective function and a plurality of decision variables, the plurality of decision variables corresponding to at least one of the resources, each of the plurality of decision variables having a respective variable value; and for a present iteration: based on a policy, selecting individual ones of the plurality of decision variables to form a subset, each of the plurality of decision variables forming the subset being configured to be updateable by a solver in a present iteration, each of the plurality of decision variables forming the subset being selected independently of another of the plurality of decision variables; determining a complement of the subset, wherein the complement and the subset are mutually exclusive, the complement being formed of other of the plurality of decision variables which are configured to be not updated by the solver in the present iteration; providing the subset and the complement to the solver, the solver being configured to optimize the scheduling model to determine respective updated variable values of the subset, each of the respective variable values of the complement being held constant by the solver in the present iteration; and if a predetermined condition is not met, using the updated variable values of the subset and the variable values of the complement as the plurality of decision variables for a subsequent iteration, wherein a subsequent subset and a subsequent complement are determined from the plurality of decision variables in the subsequent iteration.

[0008] According to one embodiment, the subset is non-mutually exclusive between the present iteration and the subsequent iteration. In one embodiment, a dimension of the subset is variable between the present iteration and the subsequent iteration. In one embodiment, prior to optimizing the scheduling model, removing the respective variable values of the

subset of decision variables, wherein the removing is a part of a destroy operation performed for the present iteration.

[0009] According to one embodiment, the subset is formed by inputting the respective variable values of the decision variables of the present iteration to a machine learning model and obtaining an output from the machine learning model. In one embodiment, the machine learning model is configured to receive a plurality of variable spaces of different dimension. The method may further comprise training the machine learning model via a reinforcement learning-based method, wherein a randomized initial variable value is provided to the reinforcement learning-based method. The method may further comprise training the machine learning model via an imitation learning-based method, wherein by use of a heuristic operator, a heuristic initial variable value is provided to the imitation learning-based method.

[0010] In one embodiment, the output is a probability distribution with a respective probability value corresponding to each of the decision variables of the present iteration, wherein each of the respective probability value corresponding to the subset is above a threshold value. In one embodiment, the method comprises performing a plurality of the subsequent iteration until the predetermined condition is met.

[0011] In one embodiment, the scheduling model is reconfigured responsive to a change in the scheduling problem. In one embodiment, at least one of the decision variables comprise one of or both of: a binary variable and a real variable. In one embodiment, at least one of the variable value is one of: an integer and a real number.

BRIEF DESCRIPTION OF DRAWINGS

[0012] Fig. 1 is a schematic diagram of a system according to an embodiment;

[0013] Fig. 2 is a data flow diagram of the system of Fig. 1 according to an embodiment;

[0014] Fig. 3 shows a schematic diagram of a policy according to an embodiment;

[0015] Fig. 4 shows a schematic diagram of a policy according to another embodiment;

[0016] Figs. 5A and 5B are data flow diagrams of a scheduling system according to an

exemplary embodiment;

[0017] Figs. 6 is a data flow diagram according to another embodiment;

[0018] Fig. 7 is an example airport ground handling schedule of an airport ground handling (AGH) system for vehicle routing problem according to an embodiment;

[0019] Figs. 8A to 8D schematically illustrate an exemplary embodiment of a policy for a scheduling model according to an embodiment;

[0020] Fig. 9 is a flowchart of a method of scheduling.

DETAILED DESCRIPTION

[0021] Reference throughout this specification to “one embodiment”, “another embodiment” or “an embodiment” (or the like) means that a particular feature, structure, or characteristic described in connection with the embodiment is included in at least one embodiment. Thus, the appearance of the phrases “in one embodiment” or “in an embodiment” or the like in various places throughout this specification are not necessarily all referring to the same embodiment. Furthermore, the described features, structures, or characteristics may be combined in any suitable manner in one or more embodiments. In the following description, numerous specific details are provided to give a thorough understanding of embodiments. One skilled in the relevant art will recognize that the various embodiments may be practiced without one or more of the specific details, or with other methods, components, materials, etc. In other instances, some or all known structures, materials, or operations may not be shown or described in detail to avoid obfuscation.

[0022] The word “exemplary” is used herein to mean “serving as an example, instance, or illustration.” Any embodiment described herein as “exemplary” is not necessarily to be construed as preferred or advantageous over other embodiments. As used herein, the singular ‘a’ and ‘an’ may be construed as including the plural “one or more” unless apparent from the context to be otherwise.

[0023] Terms such as “first” and “second” are used in the description and claims only for the sake of brevity and clarity, and do not necessarily imply a priority or order, unless

required by the context. The terms "about" and "approximately" as applied to a stated numeric value encompasses the exact value and a reasonable variance as will be understood by one of ordinary skill in the art, and the terms "generally" and "substantially" are to be understood in a similar manner, unless otherwise specified.

[0024] In the present disclosure, the terms "machine learning", "neural network", "artificial intelligence", "deep learning", "neural network" may be used interchangeably. The term "machine learning model" may generally refer to but is not limited to, one of or multiple of or a combination of: reinforced learning models, imitation learning models, supervised learning models, unsupervised learning models, regression models, classification models, neural network models, deep learning models, decision tree models, support vector machine models, naïve bayes models, clustering models, principal component analysis (PCA) models, etc.

[0025] The terms "update", "updated", "updating", and "updateable" with reference to a variable value may generally refer to the variable value being capable of being changed or altered, but is without the limitation/requirement of the actual changing or altering of variable value.

[0026] Scheduling problems are often presented in the form of a combinatorial optimization problem (COP). Typical semi-automated methods or processes make use of heuristics which are often based on trial-and-error and require input from an experienced user. Due to the large decision variables space for such COP, semi-automated optimization/solver processes are time consuming and costly. On the other hand, automated methods relying on machine learning despite having short runtimes, typically suffer from poor solution quality. Further, for large decision variables space, it is often unmanageable to train and store this many neural networks. Some conventional methods employ fixed sized sub-networks for solving the large decision variables space. However, this is unsuitable for various COPs with different numbers of variables. Further, the complexity of training such policy networks could drastically increase with the growing dimension/size of the decision variables space, since an exponential number of outputs needs to be traversed.

[0027] Fig. 1 illustrates a system 100 for a scheduling model 50 presented in the form of a combinatorial optimization problem (COP) 50, according to an embodiment of the present

disclosure. In an example of a combinatorial optimization problem 50, the COP 50 includes a plurality of variables 52, a plurality of objectives 54, and a plurality of constraints 56. Integer Program (IP) is an example of COP 50. The integer program may be defined as $\arg \min_x \{c^T x \mid Ax \leq b, x \geq 0, x \in \mathbb{Z}^n\}$, where x is a vector of n decision variables; $c \in \mathbb{R}^n$ denotes objective coefficients; the incidence matrix $A \in \mathbb{R}^{m \times n}$ and vector $b \in \mathbb{R}^m$ together define m linear constraints. With the above formulation, the size of the integer program is generally reflected by the number of variables n and constraints m . The integer program may be a mixed-integer linear programming (MILP).

[0028] In some embodiments as shown in Figs. 1 and 2, the scheduling model or COP 50 may be generally characterized or represented by a framework 200 comprising at least one objective function 210 and multiple decision variables 220 collectively known as a variable space 230 or an action space. Each of the decision variables 220 may have a respective variable value, for example a binary value of “0” or “1”. In some embodiments, the decision variables 220 may comprise at least one a binary variable and/or at least one a real variable. In some embodiments, the variable value may be an integer (e.g., -5, -10) and/or a real number (e.g., 3.5, 5.5). As an example, the framework 200 may be a Large Neighborhood Search (LNS) framework. In addition to the multiple decision variables 220, the variable space 230 may also include multiple static variables and historical variable values of the decision variables 220 both of which are configured to remain unchanged during the optimization process for the subsequent iteration. The static variables may correspond to fixed variables related to the scheduling problem which will not be optimized, such as a “refueling duration”. The historical variable values correspond to variable values of the decision variables in previous iterations. The decision variables 220 may be collectively formed as a vector, an array or a matrix depending on the framework 200. In a non-limiting example where the scheduling model relates to the scheduling of ground handling resources in an airport, the objective function 210 may correspond to reducing a downtime of all airport ground handling resources. Airport ground handling resources may include a variety and different numbers of towing vehicles, cleaning teams, fueling trucks, etc., and the decision variables 220 may correspond to the respective deployment status of each of the multiple towing vehicles, etc.

[0029] The system 100 may further include a computational device 300 for processing or

computing the decision variables 220 based on the objective function 210. The computational device 300 is configured to compute or process the decision variables 220 to obtain an optimized solution of the COP 50 according to the objective function 210. The computational device 300 may be embodied as a computer program product comprising a non-transitory computer readable storage medium having program instructions embodied therewith, the program instructions executable by a computer to cause the computer to perform computational operations. In other embodiments, the computational device 300 may include a plurality of nodes in a network for performing the computational operations.

[0030] In an embodiment, the computational device 300 may include a policy 310 and a solver 320 to perform an iterative solving/computational process. In one exemplary iteration as shown in Fig. 2, the policy 310 may be configured to segment or separate the decision variables 220, collectively known as the initial variable space 230a, into a subset 222 of decision variables 220 and a complement 223 of decision variables 220. The complement 223 of decision variables 220 being mutually exclusive to the subset 222 of decision variables 220. The subset 222 of decision variables 220 may be a vector, an array or a matrix.

[0031] In one embodiment, the policy 310 determines or selects the subset 222 of decision variables 220 from the initial variable space 230a based on the respective variable values 221 of the initial variable space 230a. In another embodiment, at least one decision variable 220 may be represented as a binary variable, i.e., as a binary variable value 221 (e.g., “1” or “0”). As it can be observed, the complement 223 of decision variables 220 is also directly or indirectly determined by the policy 310. The subset 222 of decision variables 220 and the complement 223 of decision variables 220 are mutually exclusive and collectively make up a segmented variable space 230b. Each of the decision variables 220 forming the subset 222 is configured to be updateable by a solver 320 in the present exemplary iteration. In each iteration, each of the decision variables forming the subset 222 for the present iteration is selected individually, on the basis of the policy 310 and independently of another/any other of the decision variables 220 in the initial variable space 230a. The complement 223 for the present iteration is formed by all the decision variables (in the initial variable space 230a) which are not selected for the subset 222. Each decision variable in the complement 223 is configured to be not updateable or not updated by the solver 320 in the present iteration. Advantageously, the policy 310 is agnostic or universal, and is compatible with different

solvers 320. This eliminates the need for specially designed solvers for specific applications, thus broadening the application scope of the policy 310.

[0032] Still within the same exemplary iteration, the subset 222 and the complement 223, collectively the segmented variable space 230b, are provided to a solver 320 as inputs for optimizing based on the objective function 210. The policy 310 is compatible with various solvers 320 and does not require special tuning or manual intervention for optimization. Each of the respective variable value 221 of the subset 222 is configured to be updateable or alterable, while each of the respective variable value 221 of the complement 223 is held constant. Only the variable values 221 of the subset 222 are updated to a respective updated variable value 224, while each of the variable values of the complement 223 is held constant. In doing so, a sub-optimization problem is provided for the exemplary iteration which reduces the amount of computational costs in comparison to optimizing the whole initial variable space 230a. It may be appreciated that each of the variable values 221 of the subset 222 may or may not be changed during the optimization process. However, each of the variable values 221 in the subset 222 are provided with a liberty to be updated or altered. Conversely, each of the variable values 221 in the complement 223 is held constant and hence prohibited from change or alteration for the current iteration.

[0033] As an example, a variable value which is updateable may be “1” before and after the optimization process. The same variable value may be “0” before the optimization process and “1” after the optimization process. In another example, the variable value may be “1” before the optimization process and the updated variable value may be “0” after the optimization process. Similarly, the variable value may be “1” before the optimization process and the updated variable value may also be “1” after the optimization process. In some examples, updating the variable value may cause the variable value to change from “null” before the optimization process to “0” after the optimization process.

[0034] Upon optimization, the subset 222 and the complement 223 formed a collective updated variable space 230c. The updated variable value 224 of the subset 222 and the variable values 221 of the complement 223 which corresponds to the updated variable space 230c are provided as the variable values 221 of the initial variable space 230c for a subsequent iteration 80 as illustrated in Fig. 2. That is, if a predetermined condition is not

met upon optimization in the present iteration, the updated variable values of the subset and the (not updated) variable values of the complement are used as the decision variables in the subsequent iteration 80. In each of the one or more subsequent iteration 80, the decision variables are individually selected afresh to form either one of a new subset and a new complement. That is, a subsequent subset and a subsequent complement are determined from the decision variables in the subsequent iteration. The process of obtaining the segmented variable space 230b and the updated variable space 230c are iteratively performed and halts when the predetermined condition is met. That is, the system is configured to iteratively reconfigure the subset and the complement for each iteration until the predetermined condition is met. As examples, the predetermined condition may be met when the objective function of the iteration is at a maximum/minimum, or when a number of iterations are performed. When the predetermined condition is met, the variable values 221 of the respective decision variables 220 corresponds to the current optimized solution of the scheduling model.

[0035] It may be appreciated that the respective initial variable space 230a for subsequent iterations 80 are expected to be different after undergoing the process of optimization. However, it is possible for the respective initial variable space 230a for two iterations to be the same. This is possible when the initial variable space 230a are at or nearing an optimized solution of the scheduling model or COP 50, which presents a suitable predetermined condition to halt the iterations 80.

[0036] In some embodiment, the subset 222 of decision variables 220 is non-mutually exclusive between two or more iterations 80, such as a present iteration and a subsequent iteration. In other words, the same decision variable 220 may be selected as part of the subset 222 between two or more iterations 80. This results in an overlapping of subset 222 between iterations, advantageously optimizing each decision variable 220 for multiple times, potentially avoiding local minimum/maximum issues.

[0037] In some embodiments, a dimension of the subset 222 of decision variables 220 is variable between two or more iterations 80, such as a present iteration and a subsequent iteration. In other words, the number of selected/segmented decision variables 220 in the subset 222 may be different between iterations 80. It may be advantageous to begin the

iteration process with a larger number of decision variables 220 or larger dimension of the subset 222, and progressively reduce the dimension of the subset 222 after each iteration 80 for improved computational efficiency.

[0038] In some embodiments, the policy 310 may include a machine learning model 312 configured to receive a variable space 230 and the respective variable values 221, such as the initial variable space 230a, and to provide an output 314 corresponding to the selected subset 222 of decision variables 220. The output 314 may be described as a plurality of probabilities, each of the probabilities corresponding to the probability of each respective decision variable 220 being selected to be updateable. A decision variable 220 that is selected to be updateable, i.e., allocated to the subset. The probability associated with each decision variable may be within $[0,1]$. In one example, the trained neural network may output a probability of 0.9 for one of the decision variables. This can be taken to correspond to a 90% probability that this particular decision variable will be selected to form part of a present subset 222. In the same iteration, another decision variable may be allocated a probability of 0.3 and not selected to be updateable, in which case it will not form part of the present subset 222. The respective probability is associated with one decision variable and not with a group of the decision variables. The neural network or machine learning model 312 is preferably automatically trained by training algorithms based on imitation learning or reinforcement learning. On forming the present subset 222 of updateable decision variables, the solver will optimize the COP on the basis that (i) only the updateable decision variables will be updated in the course of optimization and (ii) the other (non-updateable) decision variables are kept constant in the course of optimization. The results of the optimization by the solver may be returned to the variable space, from whence a subsequent iteration may be performed to determine the respective probabilities for each decision variables and to form a subsequent subset. The prediction of the probabilities of the decision variables being updateable and optimization based on only the selected decision variables being updateable can be repeated until the predetermined condition is met. The results of the optimization obtained are found to provide a practical way to solve the NP-hard problem. Non-limiting examples will be described in the following with more details to further aid understanding.

[0039] In an embodiment, the machine learning model or the policy network 312 may share

parameters across all variables. Therefore, the elementary policies on each decision variable are homogeneous, i.e., binary decisions. Further, patterns learned for a decision variable may be effectively reused in other decision variable and the learned policy network 312 may be applied to any problem sizes without the need of reconstructing the policy network 312. Advantageously, based on different application or availability of scheduling resources, the machine learning model 312 may be configured to receive variable spaces 230 or groups of decision variables 220 of different dimensions. For example, when there is a change in state of the scheduling model, for example a change in scheduling resources or number of constraints, the same machine learning model 312 or policy 310 may be employed to receive the corresponding variable space without a need for reconfiguration.

[0040] The policy network 312 may include mean pooling between two Multilayer Perceptrons (MLPs) with a concatenation of variables performed after the two MLPs. The concatenation of variables is then passed into a hidden layer followed by a sigmoid activation function to provide an output 314. In an example, the policy network 312 as shown in Fig. 3, given the state, i.e., a collection of features for each variable is first processed by a Multilayer Perceptron (MLP) 313 with one 128-dimensional hidden layer and Tanh activation function. The resulting 128-dimensional variable embeddings are then averaged by mean pooling 315, and further advanced by another MLP 316 with one 256-dimensional hidden layer activated by Tanh, to obtain the state embedding. Next, the global information is injected into each variable by concatenating 317 variable embeddings with the state embedding. Lastly, the state-dependent variable representations are passed through one 64-dimensional hidden layer 318 activated by Tanh, and the single-value output layer followed by sigmoid activation function. In this way, the output 314 is a probability distribution with a probability value corresponding to each of the decision variables 220 of the present iteration. The decision variables 220 may be selected as the subset 222 when the corresponding probability values are above a threshold value, for example above 0.5 or 50%. Some randomness may be added to the policy 310 to avoid stagnation and to avoid local minimum.

[0041] In some embodiments as shown in Fig. 4, the machine learning model 312 may be a graph convolutional network. The machine learning model 312 may preferably be trained via a reinforcement learning-based method. During the training of the machine learning

model 312, a randomized initial variable value 316 may be used for the reinforcement learning-based method. This advantageously facilitates machine learning model training without the need for any prior heuristic training data.

[0042] In other embodiments, machine learning model 312 may preferably be trained via an imitation learning-based method. During the training of the machine learning mode, a heuristic initial variable value 318 may be provided to the imitation learning-based method by use of a heuristic operator. In contrast to reinforcement learning-based method, imitation learning can leverage the knowledge from the existing heuristic operators, which are easy to design for routing problems, thus saving time and avoiding learning from scratch.

[0043] In other embodiments, the policy 310 may be a randomized policy, such that the subset 222 of the decision variables is selected randomly. In another embodiment, such as for ground handling operations scheduling model, selecting the vehicles which are being used is an example of heuristic initial variable value 316. Further, other decision variables which are related to the selected vehicles may also be selected. A decision variable is said to be related to a vehicle if it is related to the route of the selected vehicle.

[0044] Figs. 5A and 5B illustrate a data flow of decision variables 220 according to an embodiment of the present disclosure. For a first iteration as shown in Fig. 5A, by using a policy, an initial variable space 230a is segmented or separated 82 into a subset 222 and a complement 223, collectively referred to as the segmented variable space 230b. The selection of subset 222 may be based on the respective variable values 221 of the initial variable space 230a. In other words, the policy determines the decision variables 220 as the subset 222 by taking into account the respective variables value 221 of each decision variable 220. The respective variable values of the subset 222 are configured to be updateable while the respective variable values of the complement 223 are held constant for the current iteration. The segmented variable space 230b is provided to the solver to perform optimization 84 where only the variable values of the subset 222 are updateable. This reduces the size/dimension of the optimization problem as only the subset 222 is updateable, hence reducing the computational resources needed. During or after the optimization process, the variable values 221 of the subset 222 are updated or optimized as respective updated variable values 224 of the subset 222. For clarity, the updated variable values 224

of the subset 222 may or may not be different from the variable values 221 of the subset 224 before optimization. The updated variable values 224 of the subset 222 and the variable values 221 of the complement 223, collectively known as an updated variable space 230c, is provided to a second iteration (Fig. 5B) as an initial variable space 230c of the second iteration.

[0045] Similar to above, in the second iteration as shown in Fig. 5B, the initial variable space 230c are separated or segmented 82 by the policy into a subset 222 and a complement 223 collectively referred to as the segmented variable space 230d. It may be appreciated that the subset 222 of the first iteration and the subset 222 of the second iteration may not be mutually exclusive as shown in Figs. 5A and 5B. Further, a dimension of the subset 222 of the first iteration (e.g., 3x1 decision variables) and a dimension of the subset 222 of the second iteration (e.g., 3x2 decision variables) may not be the same. The variable values 221 of the subset 222 are then updated or optimized 84 as respective updated variable values 224. The updated variable values 224 of the subset 222 and the variable values 221 of the complement 223, collectively known as an updated variable space 230e, which is provided to a subsequent iteration 80.

[0046] In some embodiments as illustrated in Fig. 6, after separating an initial variable space 230a to obtain a segmented variable space 230b, an additional destroy operation or destroy operator 86 may be performed to remove the variable values 221 of the subset 222 of decision variables 220. The respective removed variable values 221 may then correspond to “null” value 226. In this embodiment, all of the removed variable values or “null” value 226 of subset 222 are altered or changed during optimization 86 such that no variable values remain as “null” value 226 after optimization. It may be appreciated that variable value of “null” is different from variable value of “0”. In other embodiments, some “null” value 226 may remain as the variable values 221. However, the policy 310 is configured such that decision variables 220 with “null” value 226 are not selected as part of the subset 222.

[0047] In some embodiments for a vehicle routing problem, the destroy operator 86 may be a random destroy operator configured to randomly select decision variables 220 for removing respective the variable values 221. In other embodiments, the destroy operator 86 may be a Vehicle-Random destroy operator configured to selects all the decision variables

220 of randomly chosen vehicles according to a destroy degree. In another embodiment, the destroy operator 86 may be a Vehicle-Worst destroy operator configured to select all the decision variables of vehicles with length/cost of route ranks in a topK. TopK refers to the top K elements, for example, vehicle 1, 2, 3, 4, 5 may have respective route length/distance as: 12, 31, 24, 34, 14. Therefore, topK for K=3, includes a list of vehicles [4,2,3] with the top 3 highest route length. The destroy operator 86 may be a Vehicle-Worst-Random destroy operator may be a combination of the Vehicle-Random destroy operator (50%) and Vehicle-Worst destroy operator (50%).

[0048] According to another aspect as illustrated in Fig. 7, an embodiment of an airport ground handling (AGH) system or a system for airport ground handling resources routing problem 500 is disclosed. The AGH system may be adapted from the system 100 as disclosed in the above sections. The vehicle routing problem may include scheduling a plurality of airport ground handling resources 510/520/530/540, such as service vehicles, to a plurality of aircrafts. Each of the airport ground handling resources 510/520/530/540 may correspond to at least one decision variables. The plurality of airport ground handling resources 510/520/530/540 may be grouped into a plurality of functional groups A/B/C/D, such as a plurality of fleets. Each of the plurality of functional groups A/B/C/D configured to perform a respective type of operation, with each operation including at least one related task. According to an embodiment, the vehicle routing problem may include a large number of decision variables and constraints, to the order of millions. For example, one decision variables may correspond to a movement of one of the airport ground handling resources 510/520/530/540 from one aircraft to another aircraft. In another example, one decision variable may correspond to an availability of one of the airport ground handling resources 510/520/530/540. In other example, one decision variables may correspond to a duration of service for the airport ground handling resources 510/520/530/540.

[0049] As a non-limiting example, with reference to Fig. 7, a first fleet (A) may be configured to perform the tasks of disembark A1, providing portable water A2, and unloading baggage A3. A second fleet (B) may be configured to perform the tasks of fueling B1, catering B2, cleaning B3, toilet servicing B4 and loading baggage B5. The vehicle routing problem thus constitutes a multiple-fleet vehicle routing problem with constraints of capacity, time windows and precedence.

[0050] In the example, the vehicle routing problem entails two classes of decision variables, (i) a binary variable x_{ijv}^k , which is equal to 1 if a vehicle v in fleet k travelled from flight i to j , and 0 otherwise; and (ii) a real variable T_{iv}^k , which represents the start time for servicing flight i by vehicle v in fleet k . There may be more than one objective functions, such as: aims to minimize the total cost (distance); ensures that each flight is only served by one vehicle in each fleet; ensures the continuity of the route; ensures that the number of used vehicles will not exceed the available vehicles in each fleet, etc. The vehicle routing problem may be formed as a mixed integer linear programming (MILP). on one aircraft for one airport ground handling resource 510/520/530/540.

[0051] The AGH system may include a computational device configured to schedule the plurality of functional groups to perform the respective related task for one of the aircrafts. Further, the computational device may be configured to schedule one of the functional groups A/B/C/D to perform a series of sequence related tasks. The series of sequence related tasks may also refer to precedence related tasks which require the completion of a prior related task (such as disembark A1) before commencement of a subsequent related task (providing potable water A2). The sequence related tasks may also include a predetermined order of priority among the functional groups A/B/C/D. For example, to first deploy functional group A before deploying functional group B. Alternatively, the computational device may be configured to schedule one of the plurality of functional groups A/B/C/D to perform a plurality of concurrently related tasks, such as fueling B1, catering B2 and cleaning B3 as shown in Fig. 7. For sake of clarity, concurrently related tasks may refer to overlapping related tasks having an overlap period but with different starting time and/or ending time. In some examples, the related task may be for the functional group to return to depot or to undergo maintenance, and hence is not targeted at tasks related to aircrafts.

[0052] Referring again to Fig. 7, the computational device may be configured to schedule two functional groups A/B/C/D to concurrently perform the respective related tasks. For example, the computational device may schedule the first fleet (A) and the second fleet (B) to concurrently perform their related tasks such that a downtime of the aircraft is reduced. It can be seen that the operating duration of the first fleet (A) overlaps with the operating duration of the second fleet (B). In some embodiments, the computational device may be configured to schedule a functional groups A to concurrently perform the respective related

tasks on a group of aircrafts requiring the same functional group A. In some embodiments, the computational device may be configured to dynamically solve the vehicle routing problem, responsive to an environment change, for example, a change in a number of the plurality of airport ground handling resources 510/520/530/540 and/or a number of the plurality of aircrafts. That is, the scheduling model may be configured to be responsive to a change in the scheduling problem.

[0053] The present system solves practical constraints in AGH, i.e., time windows, capacity or precedence constraints, as well as heterogeneous operations for a large volume of aircrafts. This in comparison to hand-crafted heuristics which may be intractable and is usually accompanied with substantial domain knowledge and trial-and-error efforts.

[0054] In an exemplary embodiment as illustrated in Figs. 8A to 8D, the scheduling model or COP 50 may be formulated or provided as a modified Markov Decision Process (MDP), or in other words provided as a discrete-time stochastic control process. This provides a mathematical framework for modeling decision making. The COP 50 is represented as a bipartite graph $G = (C, V, E)$, with constraints $c_i \in C$, variables $v_j \in V$ and edges $e_{ij} = (c_i, v_j) \in E$, as depicted in Fig. 8A. The policy is parameterized by a graph convolutional network (GCN), given its desirable capability in learning representation over graph-structured inputs and generalizing to different sizes. The policy first executes the graph convolution by applying two sequential message passing, i.e., from variable to constraint nodes (V to C), and from constraint to variable nodes (C to V), respectively. Accordingly, the variable and constraint embeddings are updated by the above convolutions, as shown in Fig. 8B. Since the aim is to select variables for optimization or are-optimization, the variables embeddings are aggregated by mean pooling to nodes W, each of which corresponds to a group of variables as shown in Fig. 8C. As an example, for the airport ground handling problem, the group of variables correspond to the same index of fleet and vehicle. Thus, the embedding w_h represents a group of variables related to the same vehicle in a problem instance. As depicted in Fig. 8D, the embeddings are input and processed by a MLP with a single output, followed by sigmoid activation to obtain variable selection based on probability.

[0055] The system and method as disclosed herein for solving the airport ground handling

(AGH) problem (as a non-limiting example of a practical and complex scheduling problem) is performed on various realistic scenarios. In the experiments, instances of 20 flights, 50 flights, and 100 flights, each with 10 types of operations, were first considered. As shown in Tables 1A-1C, the present method (IL-sample) was compared with a conventional IP solver CPLEX, a conventional Large Neighbourhood Search (LNS) methods (vehicle-random, vehicle-worst, vehicle-worst-random) and state-of-the-art methods for AGH. The results show that the present method outperforms all baselines. For the same runtime, the present method was able to deliver the lowest objective value and lowest gap, i.e., the best solutions of the various methods compared. The present method delivers results that is even better than that of CPLEX ($\times 5$), even though CPLEX was allowed to run 5 times longer than the present method. The versatility of the present method was also verified by adopting a different solver, i.e., OR-Tools, as shown in Table 2. As shown in Table 2, the present method achieves the best results on the large-scale AGH problem with 100 aircrafts. Notably, the system and method of the present disclosure are able to generalize well to large instances even if the policy is trained on small instances. As shown in Table 3, the scalability on larger instances (AGH200 with 200 aircrafts, formulated by 450k variables and 4000k constraints) were also verified.

Table 1A. Comparison with Baselines (AGH20)

Methods	AGH20 (6k/15k)					
	Mipgap	Limit	$\mathcal{D}_d$	Obj.	Gap	Time
CPLEX	1e-04	1m	–	86384	12.61%	1m
CPLEX ($\times 5$)	1e-04	5m	–	77630	1.65%	5m
Decomposition [10]	1e-04	45m	–	107269	41.44%	45m
GA [8]	–	1h	–	115706	51.71%	1h
Random	0.1	10s	0.4	151562	99.37%	1m
Vehicle-random	0.1	10s	0.4	86301	13.04%	1m
Vehicle-worst	0.1	10s	0.4	98383	28.31%	1m
Vehicle-worst-random	0.1	10s	0.4	90477	18.32%	1m
IL-sample (Ours)	0.1	10s	0.4	85942	12.53%	1m

Table 1B. Comparison with Baselines (AGH50)

Methods	AGH50 (30k/140k)					
	Mipgap	Limit	$\mathcal{D}_d$	Obj.	Gap	Time
CPLEX	1e-04	5m	–	365353	152.80%	5m
CPLEX ($\times 5$)	1e-04	25m	–	341397	134.65%	25m
Decomposition [10]	1e-04	47m	–	215620	47.70%	47m
GA [8]	–	1h	–	261137	77.60%	1h
Random	0.1	1m	0.4	332537	128.76%	5m
Vehicle-random	0.1	1m	0.4	166735	14.66%	5m
Vehicle-worst	0.1	1m	0.4	223397	53.43%	5m
Vehicle-worst-random	0.1	1m	0.4	170875	17.58%	5m
IL-sample (Ours)	0.1	1m	0.4	166080	14.24%	5m

Table 1C. Comparison with Baselines (AGH100)

Methods	AGH100 (120k/700k)					
	Mipgap	Limit	$\mathcal{D}_d$	Obj.	Gap	Time
CPLEX	1e-04	30m	–	696762	178.58%	30m
CPLEX ($\times 5$)	1e-04	2.5h	–	684509	173.82%	2.5h
Decomposition [10]	1e-04	57m	–	369966	48.01%	57m
GA [8]	–	1h	–	438542	76.74%	1h
Random	0.2	10m	0.4	491285	96.42%	30m
Vehicle-random	0.2	10m	0.4	311035	24.21%	30m
Vehicle-worst	0.2	10m	0.4	663911	164.62%	30m
Vehicle-worst-random	0.2	10m	0.4	316498	26.31%	30m
IL-sample (Ours)	0.2	10m	0.4	302512	20.86%	30m

Table 2. Results with OR-Tools

Methods	AGH20		AGH50		AGH100	
	Obj.	Gap	Obj.	Gap	Obj.	Gap
OR-Tools	80073	4.07%	150835	3.12%	323762	30.47%
Random	135183	75.86%	261293	78.77%	466692	88.29%
Vehicle-random	88205	14.34%	159680	9.15%	313815	26.63%
Vehicle-worst	88413	14.85%	184750	26.33%	502614	102.82%
Vehicle-worst-random	88208	14.53%	162148	10.81%	352973	42.20%
IL-sample(Ours)	86479	12.27%	151488	3.59%	300160	20.97%

Table 3. Results on AGH200 (450k/4000k)

Method	Mipgap	Timelimit	Destroy degree	Obj.	Gap	Time
CPLEX	1e-04	2h	–	1322757	152.72%	2h
Decomposition	1e-04	2.2h	–	663254	26.73%	2.2h
Random	0.2	30m	0.2	1321166	152.41%	2h
Vehicle-random	0.2	30m	0.2	666801	27.33%	2h
Vehicle-worst	0.2	30m	0.2	1080257	106.66%	2h
Vehicle-worst-random	0.2	30m	0.2	741957	41.27%	2h
IL-sample (Ours)	0.2	30m	0.2	652405	24.59%	2h
OR-Tools	–	2h	–	1288353	146.28%	2h
Random	–	30m	0.2	1318479	151.97%	2h
Vehicle-random	–	30m	0.2	1084422	107.24%	2h
Vehicle-worst	–	30m	0.2	1120329	114.09%	2h
Vehicle-worst-random	–	30m	0.2	1087186	107.76%	2h
IL-sample (Ours)	–	30m	0.2	1015181	94.05%	2h

[0056] According to another aspect, a method 700 of scheduling resources is disclosed. The method 700 includes: configuring a scheduling model comprising at least one objective function and multiple decision variables, each of the decision variables having a respective variable value (710); and for a present iteration: based on a policy, selecting individual ones of the plurality of decision variables to form a subset (720); determining a complement of the subset, wherein the complement and the subset are mutually exclusive (730); providing the subset and complement to a solver to optimize the scheduling model to determine a respective updated variable value of the subset, wherein for the present iteration, each of the respective variable value of the subset is updateable by the solver and each of the respective variable value of the complement is held constant (740); and providing the updated variable values of the subset and the variable values of the complement as the plurality of the decision variables for a subsequent iteration (750).

[0057] The method 700 may further include prior to optimizing the scheduling model, removing the variable values of the subset of decision variables. The method 700 may also include inputting the respective variable values of the present iteration to a machine learning model and obtaining an output from the machine learning model. The method 700 may

further include training the machine learning model via a reinforcement learning-based method, and optionally, providing a randomized initial variable value to the reinforcement learning-based method. The method 700 may further include training the machine learning model via an imitation learning-based method, and optionally, by use of a heuristic operator, providing a heuristic initial variable value to the imitation learning-based method.

[0058] The foregoing describes various examples of a system and a method of scheduling, presented in a combinatorial optimization problem (COP) form. The method may be described as including iteratively performing the steps of: factorizing or segmenting the large decision variable space (or action space) into a subset and a complement using a policy, and optimizing the COP using a solver allowing the subset to be updateable while fixing the complement. The method may optimize the initial solution and obtain a solution that is close to the optimum in a shorter runtime compared to conventional solvers. It may be appreciated that the present method of configuring a subset of the decision variable space is a departure from the conventional Markov decision process (MDP). The conventional MDP mandates the formation of fixed subsets, i.e., the way in which variables are grouped into subsets is fixed throughout all iterations of an optimization routine. The members of the subset of the present method are individually selected at each iteration such that the constitution of the subset may change from iteration to iteration. One of ordinary skill in the art would appreciate that the present method is a departure from the conventional MDP, and is likely to consider working with individual variables as being less efficient than working with fixed subsets. The experimental data surprisingly demonstrated that the present method can greatly reduce computational costs, while allowing a balance between computational resources and solution quality. In comparison with traditional solvers, the present method and system can reach a superior solution in a shorter time period. The reduction in computational time is especially pronounced for large-scale problems, such as airport ground handling problem for 100 or 200 flights, such that a re-scheduling can be performed within a few hours instead of a few days/weeks. The present method can also be used in other resource allocation problems involving high complexity.

[0059] Another advantage of the present method and system is that it can be integrated with any of known or new solvers to deliver improved performance. The policy is configured to share parameters across all variables/dimensions. This allows for the elementary policies on

each variable/dimension to be homogeneous, i.e., to be formulated as binary decisions. Advantageously, the patterns learned for a variable/dimension can effectively be reused in other variable/dimensions. The policy may be applied to any scheduling model sizes without the need for reconstruction of the policy. In other words, the present method and system provides a universal policy that can be applied in various different applications, and is not limited to an airport ground handling resource allocation/scheduling COP.

[0060] All examples described herein, whether of apparatus, methods, materials, or products, are presented for the purpose of illustration and to aid understanding, and are not intended to be limiting or exhaustive. Various changes and modifications may be made by one of ordinary skill in the art without departing from the scope of the invention as claimed.

CLAIMS

1. A system for scheduling a plurality of airport ground handling resources, the system comprising:

a scheduling model configured with a plurality of decision variables, the plurality of decision variables corresponding to at least one of the plurality of airport ground handling resources, each of the plurality of decision variables having a respective variable value; and

a computational device, the computational device being configured to perform a method of:

for a present iteration and based on a policy, selecting individual ones of the plurality of decision variables to form a subset, each of the plurality of decision variables forming the subset being configured to be updateable by a solver in a present iteration, each of the plurality of decision variables forming the subset being selected independently of another of the plurality of decision variables;

determining a complement of the subset, wherein the complement and the subset are mutually exclusive, the complement being formed of other of the plurality of decision variables which are configured to be not updated by the solver in the present iteration;

providing the subset and the complement to the solver as inputs, the solver being configured to optimize the scheduling model to determine respective updated variable values of the subset in the present iteration, each of the respective variable values of the complement being held constant by the solver in the present iteration; and

if a predetermined condition is not met, using the updated variable values of the subset and the variable values of the complement as the plurality of decision variables for a subsequent iteration, wherein a subsequent subset and a subsequent complement are determined from the plurality of decision variables in the subsequent iteration.

2. The system as recited in claim 1, wherein the subset is non-mutually exclusive between the present iteration and the subsequent iteration.
3. The system as recited in claim 1 or claim 2, wherein a dimension of the subset is variable between the present iteration and the subsequent iteration.
4. The system as recited in any one of claims 1 to 3, wherein the method further comprises prior to optimizing the scheduling model, removing the respective variable values of the subset for the present iteration as a part of a destroy operation.
5. The system as recited in any one of claims 1 to 4, wherein the subset is formed by inputting the respective variable values of the present iteration to a machine learning model and obtaining an output from the machine learning model.
6. The system as recited in claim 5, wherein the machine learning model is trained via an imitation learning-based method, wherein by use of a heuristic operator, a heuristic initial variable value is provided to the imitation learning-based method.
7. The system as recited in claim 5, wherein the machine learning model is trained via a reinforcement learning-based method, and wherein a randomized initial variable value is provided to the reinforcement learning-based method.
8. The system as recited in any one of claims 5 to 7, wherein the output is a probability distribution with a probability value corresponding to each of the decision variables of the present iteration, wherein each of the respective probability value corresponding to the subset is above a threshold value.
9. The system as recited in any one of claims 1 to 8, wherein the system is configured perform a plurality of the subsequent iteration until the predetermined condition is met.
10. The system as recited in any one of claims 1 to 9, wherein the plurality of airport ground handling resources is grouped into a plurality of functional groups, each of the plurality of

functional groups being configured to perform at least one respective related task, and wherein the computational device is configured to schedule the plurality of functional groups to perform the at least one respective related task.

11. The system as recited in claim 10, wherein the computational device is configured to schedule at least one of the plurality of functional groups to perform a plurality of related tasks, wherein the plurality of related tasks being a series of sequence related tasks.

12. The system as recited in claim 10, wherein the computational device is configured to schedule two or more of the plurality of functional groups to concurrently perform the at least one respective related tasks.

13. The system as recited in any of claims 10 to 12, wherein one of the decision variables corresponds to at least one of:

- a movement of one of the plurality of airport ground handling resources from one aircraft to another aircraft;

- an availability of one of the airport ground handling resources; and

- a duration of service for the one aircraft for one of the plurality of airport ground handling resources.

14. A method of scheduling resources, the method comprising:

- providing a scheduling model configured with at least one objective function and a plurality of decision variables, the plurality of decision variables corresponding to at least one of the resources, each of the plurality of decision variables having a respective variable value; and

- for a present iteration:

- based on a policy, selecting individual ones of the plurality of decision variables to form a subset, each of the plurality of decision variables forming the subset being configured to be updateable by a solver in a present iteration, each of the plurality of decision variables forming the subset being selected independently of another of the plurality of decision variables;

determining a complement of the subset, wherein the complement and the subset are mutually exclusive, the complement being formed of other of the plurality of decision variables which are configured to be not updated by the solver in the present iteration;

providing the subset and the complement to the solver, the solver being configured to optimize the scheduling model to determine respective updated variable values of the subset, each of the respective variable values of the complement being held constant by the solver in the present iteration; and

if a predetermined condition is not met, using the updated variable values of the subset and the variable values of the complement as the plurality of decision variables for a subsequent iteration, wherein a subsequent subset and a subsequent complement are determined from the plurality of decision variables in the subsequent iteration.

15. The method as recited in claim 14, wherein the subset is non-mutually exclusive between the present iteration and the subsequent iteration.

16. The method as recited in any of claims 14 to 15, wherein a dimension of the subset is variable between the present iteration and the subsequent iteration.

17. The method as recited in any of claims 14 to 16, further comprising: prior to optimizing the scheduling model, removing the respective variable values of the subset of decision variables, wherein the removing is a part of a destroy operation performed for the present iteration.

18. The method as recited in any of claims 14 to 17, wherein the subset is formed by inputting the respective variable values of the decision variables of the present iteration to a machine learning model and obtaining an output from the machine learning model.

19. The method as recited in claim 18, wherein the machine learning model is configured to receive a plurality of variable spaces of different dimension.

20. The method as recited in claim 18 or claim 19, further comprising training the machine learning model via a reinforcement learning-based method, wherein a randomized initial variable value is provided to the reinforcement learning-based method.

21. The method as recited in claim 18 or claim 19, further comprising training the machine learning model via an imitation learning-based method, wherein by use of a heuristic operator, a heuristic initial variable value is provided to the imitation learning-based method.

22. The method as recited in any one of claims 18 to 21, wherein the output is a probability distribution with a respective probability value corresponding to each of the decision variables of the present iteration, wherein each of the respective probability value corresponding to the subset is above a threshold value.

23. The method as recited in any one of claims 14 to 22, wherein the method comprises performing a plurality of iterations until the predetermined condition is met.

24. The method as recited in any one of claims 14 to 23, wherein at least one of the decision variables comprise one of or both of: a binary variable and a real variable.

25. The method as recited in any one of claims 14 to 23, wherein at least one of the variable value is one of: an integer and a real number.