Volume 3, Issue 12, December 2013 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

A Decision Support System Based on Record to Record Travel Metaheuristic for Preemptive Continuous Problems

Souhail Dhouib
Associate Professor of Management Information System,
Faculty of Administrative and Financial Sciences,
Al- Baha University, Kingdom Saudi Arabia

Abstract— In this paper, a Decision Support System based on Record to Record Travel metaheuristic (named DSS-RRT) is developed to solve preemptive continuous problems. The record to record travel metaheuristic is used for the model subsystem. The 4th Dimension language is used to develop interactive and convivial dialog subsystem. Three engineering design problems represent the data subsystem. Experiments on literature problems show that the proposed RRT_DSS is more effective.

Keywords— Decision support system, metaheuristic, record to record travel algorithm, Optimisation, continuous problems.

I. INTRODUCTION

A Decision Support System (DSS) is a computer centred system in which three components are integrated to support structured or semi-structured decision making in an interactive environment. The first component is the data subsystem that offers relevant data in the desired format to the decision-maker. The second one is the model subsystem analyses and interprets data through several decision models. The third component is the dialog subsystem which provides an interactive interface between the decision-maker and the system (see [1, 4, 12, 13, 14, 20, 22, 23 and 24]). The objective of a model subsystem is to model the decision environment; the modelling tool should be simple to use and flexible that's why a metaheuristic method is used: the RRT. This metaheuristic is derived from the Simulated Annealing (SA) algorithm.

The data subsystem provides the input data to the modelling subsystem. Three engineering design problems are used. The dialog subsystem offers an interactive interface with the decision-maker. The proposed DSS-RRT works in three phases: In the first phase, it individually minimizes the two criteria using two agents. Each agent starts from a random initial solution, minimizes a specific criterion by applying RRT and stops after I iterations. In the second phase, the decision maker fixes a goal for each criterion based on the minimum found in the first phase. In the third phase, DSS-RRT framework launches two agents with the adaptive TCM memory. Each agent uses the RRT metaheuristic with a random initial solution and cooperates with other agents throw the adaptive TCM memory. The TCM is composed of two lists. The list1 is used to archive the blocked solutions (local minima). The list2 serves to save worst transformations when searching a neighbourhood solution. We note that the first phase has been executed once, but the second and the third phases have been repeated until the decision maker will be satisfied with the result found by the algorithm. The originality of the proposed DSS is to hierarchically minimize the deviation from the goals fixed in the second phase (not to the optimal). If the goal is equal to the minimum found by each agent in the first phase, the problem will be a lexicographic goal programming one. Our aim is to provide a Decision Support System (DSS) to interactively solve preemptive multi-criteria scheduling problems with goals. For that, a multi-agents framework based on the Record to Record Travel (RRT) metaheuristic combined with an adaptive memory named Taboo Central Memory (TCM) is developed. This approach is named Record to Record Travel Decision Support System (DSS-RRT).

The next section describes the RRT metaheuristic. Section 3 presents the TCM memory. Section 4 describes the proposed DSS-RRT framework which solves interactively and hierarchically two criteria. Finally, in section 5, the DSS-RRT is tested to solve three engineering design problems.

II. THE RECORD TO RECORD TRAVEL ALGORITHM (RRT)

The SA is a general local search algorithm, introduced by [4]. This algorithm is analogous to the physical annealing of solids. SA starts from an initial solution at a high temperature. Then, it attempts to move from the current solution to one of its neighbourhood solutions according to annealing schedule. At each move, the objective function value (Δ) is computed. If the new solution has a best objective function value, it is accepted. However, if the new solution yields bad value, it is accepted according to the acceptance probability, which is defined as $\exp(-\Delta/T)$, where T is the current temperature. This probability is compared with a number randomly selected from (0, 1). If r is less than or equal to the probability value, the new configuration is accepted; otherwise, it is rejected. This process is repeated L iterations at each temperature to reach the thermal equilibrium, where L is a control parameter, usually called the Markov chain length.

The value of *T* is gradually decreased by using a cooling schedule until attaining the stopping criterion. The algorithm is stopped when the optimal solution is attained or a pre-established temperature value is reached. SA is capable of jumping out of the local optima for global optimization, by accepting bad solutions, and gradually converges as the temperature falls to zero.

The SA algorithm requires the calibration of the initial temperature, the length of the Markov chain and the cooling coefficient. In [4], Dueck developed some derived algorithms named threshold algorithms (standard threshold accepting algorithm, great deluge algorithm and RRT), in order to simplify the SA algorithm. These algorithms resemble in their structure to the SA method. The differences lie in their acceptance rules for bad intermediate solutions. In the SA, this decision is taken by referring to the principle of annealing thermodynamic, whereas the derived algorithm is taken by a deterministic way. The RRT algorithm is structured even simpler than the SA algorithm. It is not necessary to compute probabilities or to make random decisions. It has the advantage that it depends only on small number (one single) of parameters. In [4], Dueck claimed that this algorithm provides better results than SA algorithm. For this reasons, this derived algorithm, named record-to-record travel, is utilized in this paper. The most important difference between those methods and the SA is how to accept the worse intermediate solution. In the SA, this decision is taken by referring to the principle of annealing thermodynamic, whereas in the derived algorithms, is by a deterministic way. The essential difference between SA and its derivate consists of the different acceptance rules. SA accepts worse solutions only with probabilities, standard threshold accepts every new configuration which is not much worse than the old one, great deluge algorithm accepts every solution less than the water level and RRT accepts every solution which is not much worse than the better one.

The RRT algorithm starts from an initial solution randomly created. Then, it attempts to move from this current solution to a new one that is generated by introducing a small change. At each move, the objective functions of these two solutions are compared. If the new solution (*s'*) has the best value, it is accepted, and it serves as the current solution for the next step. However, the objective function of this current solution and the better one (*s**) are compared (Δ := Cost(*s'*) - Cost(*s**)). If Δ is smaller than the maximum deterioration parameter (*max-det*), accept it. Else, generate another neighbourhood solution and repeat the process. In this algorithm, all solutions, far from the best solution *s**, are rejected. Thus, a fixed value of maximum deterioration (*max-det*) is defined. It does not vary in the execution process.

The general body of the RRT algorithm:

1. Fixed value of *max – det*
2. Select randomly an initial solution *s*
3. Evaluate *s*
4. Set *s** := *s*
5. Repeat
6. Select at random a new solution *s'* in the neighbor of *s*
7. Evaluate *s'*
8. Δ := *Cost*(*s'*) - *Cost*(*s**)
9. If $\Delta \leq$ *ma –det* then
10. Set *s* := *s'* /*accept every solution which is not much worst than the best one */
11. If *s* < *s** then
12. Set *s** := *s*
13. End if
14. End if
15. Until stopping-criterion is satisfied
16. Return *s**

III. TABOO CENTRAL MEMORY (TCM)

The Record to Record Travel (RRT) algorithm has no memory. We have sought, on the one hand, to make it more intelligent by starting from different initial solutions in several agents, on the other hand, to make it aware of its past by archiving information in an adaptive memory, called TCM, composed of two lists. This memory is inspired from taboo search algorithm's list. We proposed to add a second list, this one quickly improves the quality of solutions. The first list (*list1*) contains several local minima obtained solutions within P iterations. Thus, if the current solution is taboo, we know that it is a zone visited before, and that we will waste time trying to refine it without improvement. For that, the algorithm generates transformations on the current solution to generate a new one. We put in the second list (*list2*) of the TCM the transformations which do not improve the current solution (when choosing neighborhoods). The TCM plays thus a double role. First, it centralizes the agent's communication (eliminates direct message's transfer between agents). Second, it cooperates between agents to prohibit repeating the treatment of another agent, even partially.

We can use several techniques to formalize this taboo memory. In order to prohibit the system's saturation with a TCM heavily-sized, FIFO technique (First In First Out) is used.

IV. THE RECORD TO RECORD TRAVEL DECISION SUPPORT SYSTEM (DSS-RRT)

The proposed DSS-RRT framework is composed of three phases. In the first phase, it starts by randomly creating two different solutions. Then, it assigns to each agent an initial solution. Every one applies RRT algorithm without communicating with the other agents and gives the minimum criterion's value. The first phase will be stopped after a fixed number (*I*) of iterations, which allows the launch of the second phase.

The second phase shows the DSS-RRT interactivity. Each agent presents the best solution found for its criterion. These best solutions proposed by agents serve to fix a goal for each criterion. In fact, the decision maker has an idea about every criterion's solution, so he can specify a goal to each criterion. In the third phase, two agents are launched and the TCM lists are initialized. Each agent activates the RRT algorithm with a random initial solution. Each agent improves its current solution with a small change and cooperates with the other agents by the adaptive memory TCM. The agent proceeds with a new change on the current solution and inserts the worst transformation in the TCM's *list2*. After *P* iterations the best solution (local minimum) found is stocked in the TCM's *list1*.

Only in phase three, agents communicate with the TCM memory about all new solutions generated and all bad transformations on each current solution. Suppose that the new solution belongs to the *list1*, this algorithm makes the solution taboo and generates a kick. As well, if the modification on the current solution belongs to the *list2*, this algorithm applies another modification on the current solution to generate a new one.

The interactivity of this approach resides in repeating phases 2 and 3 until the decision maker will be satisfied.

The proposed DSS-RRT framework is composed of three phases.

- **Phase 1: Search for the best solution individually for each criterion**
  Select N_c criteria in the desired order, create N_c agents, affect one criterion to each agent, each one optimizes individually its criterion by the RRT algorithm to determines the minimal value, z_c ($c = 1,...,N_c$).
- **Phase 2: Fixing goals**
  Fix the goals $g_c >= z_c$.
- **Phase 3: Sophistication of the solution**
  Step 0: initialize the two TCM lists.
  Step 1: Initialize the RRT parameters.
  Step 2: Perform, on solution *X* an insertion (which doesn't belong to *list2*), to obtain a neighbor solution *X'* (which doesn't exist in *list1*). $c \leftarrow 1$ (*c* is the criterion's number). Calculate *dc*, the solution's deviation from goal ($d_c \leftarrow |F(X) - g_c|$).
  Step 3: Launch the 2 agents from different initial solutions
  Case 1: $d_c(X') < d_c(X)$
  The solution is accepted. $X_{n+1} \leftarrow X'$ If $d_c(X') \le d_c(\hat{X}) : \hat{X} \leftarrow X'$.
  Case 2: $d_c(X') > d_c(X)$
  The solution is accepted according to RRT criterion based on water level value.
  Case 3: $d_c(X') = d_c(X)$
  $c \leftarrow c+1$ (Consider the next criterion) If $c \neq N_c$,
  Return to Case 1.
  Step 4: If the number of iterations is reached, reduce the water level and return to Step 2.
  Step 5: If the stopping criterion is reached, go to step 6. Else return to Step 2.
  Step 6: Stop.

After achieving this phase and if the results found are not interesting, the decision-maker decides to reactive phase 2 and phase 3 until the algorithm gives good solutions.

V. EXPERIMENTAL RESULTS

In this section, three engineering design problems are used to evaluate the proposed DSS-RRT framework. This framework is implemented in 4th Dimension language and run under the WINDOWS XP operating system on a DELL OPTIPEX GX 520 Personal Computer.

A. Problem 1

A lexicographic goal programming problems with continuous variables is presented and solved in [3] by SA algorithm. Also this problem is solved with taboo search algorithm and using simplex based algorithm. This problem is chosen to prove our RRT_DSS platform performance to solve lexicographic continues goal programming problems.

Lexmin $\{ z_1 = (d_1^- + d_1^+), z_2 = (d_2^- + d_2^+), z_3 = (d_3^- + d_3^+) \}$

$$7x_1 + 3x_2 + d_1^- - d_1^+ = 40$$

$$10x_1 + 5x_2 + d_2^- - d_2^+ = 60$$

$$5x_1 + 4x_2 + d_3^- - d_3^+ = 35$$

$$100x_1 + 60x_2 \le 600$$

$$x_1, x_2, d_1^-, d_1^+, d_2^-, d_2^+, d_3^-, d_3^+ \ge 0$$

The optimal solution of the problem is given by $x_1 = 6$, $x_2 = 0$, $z_1 = 2$, $z_2 = 0$, $z_3 = 5$.

Our algorithm randomly starts from an initial feasible solution, moves from a current solution to a neighbour and the *integer variables* are chosen with a *step_size* $_{x1,x2}$ =(5,5): $x_i^* = x_i$ + integer [(2*random() – 1) * *step_size* $_{x1,x2}$

The proposed RRT_DSS platform found the optimal solution in less than one second.

B. Problem 2: The supported I-beam

This problem is chosen to prove our RRT_DSS's platform performance to solve continuous lexicographic goal programming problems. This example is a design optimization problem for a supported I-beam. Osyczka in [18] originally modelled this problem. Coello and Christiansen in [6] remodelled the problem as a multiple objective optimization problem and solved the model using their genetic algorithm, which is known as MOSES. Their model is also presented as a preemptive goal program and is solved by Baykasoğlu in [2] using SA algorithm.

The mathematical model is given as follows.

$$lex\min\left\{z_1=\left(d_1^+\right),z_2=\left(d_2^-\right)\right\}$$

Subject to:

$$\left(2x_2x_4+x_3\left(x_1-2x_4\right)\right)-d_1^+=127.46$$

$$\frac{60000}{x_3\left(x_1-2x_4\right)^3+2x_2x_4\left(4x_4^2+3x_1\left(x_1-2x_4\right)\right)}-d_2^+=0.0059$$

$$16-\frac{180000x_1}{x_3\left(x_1-2x_4\right)^3+2x_2x_4\left(4x_4^2+3x_1\left(x_1-2x_4\right)\right)}-\frac{15000x_2}{\left(x_1-2x_4\right)x_3^3+2x_4x_2^3}\geq 0,$$

$$10\leq x_1\leq 80,\ 10\leq x_2\leq 50,\ 0.9\leq x_3\leq 5,\ 0.9\leq x_4\leq 5$$

$x_1, x_2, x_3, x_4 \geq 0$ (*continuous*).

The proposed RRT_DSS algorithm converged to the following solution in 3 seconds of computational time: $x_1=50.8180$, $x_2=38.7575$, $x_3=0.9132$, $x_4=1.1800$ with $z_1=8.263213$ and $z_2=0.071007$.

Our algorithm randomly starts from an initial feasible solution and then uses theses parameters: To move from a current solution to a neighbour, the *integer variables* are chosen with a *step* $c_{x_1,x_2,x_3,x_4}=(3,3,0.75,0.75)$.

As shown in table 1, this solution is better than the solution of Baykasoğlu [2, 3] with a less number of parameters and computational time (ACT is the computation time). The proposed RRT_DSS found a better value for the first objective z_1 (48.55% better than the solution of Baykasoğlu in [3]) in 3 secondes.

TABLE I

COMPARISON OF THE BEST RESULTS FOR EACH OBJECTIVE OF TEST SUPPORTED I-BEAM PROBLEM

Techniques	**Z1**	**Z2**	**ACT**
SA	16.0607	0.031863	9
RRT_DSS	8.26321	0.071007	3

This solution is better than the solution found by Baykasoğlu in [3] because it is a lexicographic problem with z_1 in the first priority order and z_2 in the second priority order. Also the proposed RRT_DSS converges to this solution in a less computational time (3 seconds).

C. Problem 3: The Machine tool spindle

This is a design optimization problem for a Machine tool spindle, it was originally presented by [8] and solved by [6] through a genetic algorithm (MOSES). As well, the Simulated Annealing algorithm (SA) is used to solve this problem by [3] and the taboo search algorithm (MOTS) was also applied to solve it by [2].

The model is shown as follows:

$$lex\min\left\{z_1=\left(\delta_1^+\right),z_2=\left(\delta_2^-\right)\right\}$$

Subject to:

$$\left(\frac{\pi}{4}\left[a\left(d_a^2-d_0^2\right)+l\left(d_b^2-d_0^2\right)\right]\right)-\delta_1^+=450000,$$

$$\left(\frac{Fa^3}{3EI_a}\left(1+\frac{l}{a}\frac{I_a}{I_b}\right)+\frac{F}{c_a}\left[\left(1+\frac{a}{l}\right)^2+\frac{c_aa^2}{c_bl^2}\right]\right)-\delta_1^+=0.011,$$

$$l-l_g\leq 0,\ l_g-l\leq 0,\ d_{a_1}-d_a\leq 0,\ d_a-d_{a_2}\leq 0,\ d_{b_1}-d_b\leq 0,$$

$$d_b-d_{b_2}\leq 0,\ d_{0m}-d_0\leq 0,\ p_1d_0-d_b\leq 0,\ p_2d_b-d_a\leq 0,$$

$$\left|\Delta_a + (\Delta_a - \Delta_b)\frac{a}{l}\right| - \Delta \le 0,$$

$$I_a = 0.049\left(d_a^4 - d_0^4\right),$$

$$I_b = 0.049\left(d_b^4 - d_0^4\right), c_a = 35400\left|\delta_{ra}\right|^{1/9} d_a^{10/9}, c_b = 35400\left|\delta_{rb}\right|^{1/9} d_b^{10/9},$$

$d_0, l \ge 0,$ are continuous and d_a, d_b are discrete.

In the earlier model, d_a and d_b are discrete variables; d_a should be selected from the following set {80, 85, 90, 95}, and d_b from the set {75, 80, 85, 90}.

The TA-MA algorithm converged to the following best solution in 2 seconds of computational time: $d_0 = 27$, $d_b = 95$, $d_b = 80$, $l = 195.827$ with $z_1 = 474655.12$, $z_2 = 0.03718$.

TABLE IIIII
RESULTS FOR THE MACHINE TOOL SPINDLE PROBLEM

Techniques	**Z1**	**Z2**	**ACT**
Genetic Algorithm (MOSES)	494 015.44	0.03808	-
SA	499 182.12	0.03779	11 s
Taboo Search (MOTS)	497 644.10	0.03783	9 s
RRT_DSS	474655.12	0.03718	1 s

As shown in table 2, the solution found by the DSS-RRT is the best one with a less number of parameters and computational time (1 second).

VI. CONCLUSIONS

A Decision Support System based on Record to Record Travel metaheuristic (DSS-RRT) is proposed to solve continuous preemptive multi-criteria problems. The computational result shows that presented DSS-RRT improves the best known results by a significant margin, and outperforms taboo search algorithm and genetic algorithm. Due to the robust performance of the proposed DSS-RRT, further research will be made to check its performance on multiple objective combinatorial optimisation problems to find Pareto optimal set solutions.

REFERENCES

[1] M. J. Alves, and J. Clımaco, *A note on a decision support system for multiobjective integer and mixed-integer programming problems*, European Journal of Operational Research, vol 155, pp 258–265, 2004.
[2] A. Baykasoğlu, *Goal programming using multiple objective tabu search*, Journal of Operational Research Society, Vol. 52, pp. 1359–1369, 2001.
[3] A. Baykasoğlu, *Preemptive goal programming using simulated annealing*, Engineering Optimization, Vol. 37 (1), pp. 49–63, 2005.
[4] P. Caricato and A. Grieco, *A DSS for production planning focused on customer service and technological aspects*, Robotics and Computer-Integrated Manufacturing, vol 25, pp 871–878, 2009.
[5] H.A.J. Crauwels, C.N. Potts and V. Wassenhove, *Local search heuristics for the single machine total weighted tardiness scheduling problem*, INFORMS Journal on computing, vol 10, pp 341–350, 1997.
[6] C. Coello and A.D. Christiansen, *MOSES: a multiple objective optimization tool for engineering design*, Engineering Optimization, Vol. 31, pp. 337–368, 1999.
[7] G. Dueck, *New optimization heuristics: the record to record travel algorithm and the record-to-record travel*, Journal of Computational Physics, vol 104, pp 86–92, 1993.
[8] H. Eschenauer, J. Koski and A. Osyczka, *Multicriteria Design. Optimization*, Springer-Verlag, 1990.
[9] A.K. Gupta and A.I. Sivakumar, *Multi-objective scheduling of two-job families on a single machine*, Omega, vol 33, pp 399–405, 2005.
[10] J. N. Gupta, D. Neppalli, R. Venkata and F. Werner, *Minimizing total flow time in a two-machine flowshop problem with minimum makespan*, International Journal of Production Economics, vol 69, Issue 3, pp 323–338, 2001.
[11] J. Gupta and J. C. Ho, *Minimizing makespan subject to minimum flowtime on two identical parallel machines*, Computers & Operations Research, vol 28, pp 705–717, 2001.
[12] R. Klashner and S. Sabet, *A DSS Design Model for complex problems: Lessons from mission critical infrastructure*, Decision Support Systems, vol 43, pp 990–1013, 2007.
[13] R. Kohli and S. Devaraj, *Contribution of institutional DSS to organizational performance: evidence from a longitudinal study*, Decision Support Systems vol 37, pp 103–118, 2004.

[14] R. B. Lopes, S. Barreto, C. Ferreira and B. R. Santos, *A decision-support tool for a capacitated location-routing problem*, Decision Support Systems, vol 46, pp 366–375, 2008.
[15] T. Loukil,, J. Teghem and D. Tuyttens, *Solving multi-objective production scheduling problems using metaheuristics*, European Journal of Operational Research, vol 161, pp 42–61, 2005.
[16] E. J. Mendoza, A. L. Medaglia and N. Velasco, *An evolutionary-based decision support system for vehicle routing: The case of a public utility*, Decision Support Systems, vol 46, pp 730–742, 2009.
[17] A. Nagar, J. Haddock and S. Heragu, *Multiple and bicriteria scheduling: A literature survey*, European Journal of Operational Research, vol 81, pp 88–104, 1995.
[18] A. Osyczka, *Multicriteria optimization for engineering design: design optimization*, edited by J.S. Gero, (Academic Press), pp. 193–227, 1985.
[19] B. Roy, *Robustness in operational research and decision aiding: A multi-faceted issue*, European Journal of Operational Research, vol 200, pp 629–638, 2010.
[20] R. Ruiz, C. Maroto and J. Alcaraz, *A decision support system for a real vehicle routing problem*, European Journal of Operational Research, vol 153, pp 593–606, 2004.
[21] J.G. Shanthikumar, *Scheduling n jobs on one machine to minimize the maximum tardiness with minimum number tardy*, Computer and operations research, vol 10, N3, pp 255–266, 1983.
[22] H. C. Siew, *The roles of user motivation to perform a task and decision support system (DSS) effectiveness and efficiency in DSS use*, Computers in Human Behavior, vol 25, pp 217–228, 2009.
[23] R. Soncini-Sessa, A. Castelletti and E. Weber, *A DSS for planning and managing water reservoir systems*, Environmental Modelling & Software vol 18, pp 395–404, 2003.
[24] L. Tang and G. Wang, *Decision support system for the batching problems of steelmaking and continuous-casting production*, Omega, vol 36, pp 976–991, 2008.

