- Word Count: 5504

Plagiarism Percentage

8%

sources:

1 3% match (Internet from 22-Sep-2015)
http://140.118.5.71/sos/

2 2% match (Internet from 05-Jan-2018)
https://www.scipedia.com/search?q=Differential+evolution&search_type=tags

3 2% match (Internet from 09-Sep-2017)
http://www.ijeit.com/Vol%206/Issue%2012/IJEIT1412201706_04.pdf

paper text:

Optimizing Multiple Resources Leveling in the Multiple Projects using Discrete Symbiotic Organisms Search 1

Min-Yuan Cheng, Doddy Prayogo, Duc-Hoc Tran 1

Abstract

Resource leveling is the process used within project scheduling to reduce fluctuations in resource usage over the period of project implementation. These fluctuations frequently create the untenable requirement of regularly hiring and firing temporary staff resources to meet short-term project needs. Construction project decision makers currently rely on experience- based methods to manage fluctuations. However, these methods lack consistency and may result in unnecessary wastage of resources or costly schedule overruns. This research introduces a novel 2

Discrete Symbiotic Organisms Search for optimizing multiple resources leveling in the multiple projects scheduling problem (DSOS-MRLMP). The optimization model is proposed based on a recently developed metaheuristic algorithm, the Symbiotic Organisms Search (SOS) algorithm. SOS mimics the symbiotic

relationship strategies that organisms use to survive in the ecosystem. Experiments results and statistically test

indicate that the proposed model obtains optimal results more reliably and efficiently than the other optimization **algorithms considered. The proposed optimization model is a promising alternative approach to assist project managers to handle resource-leveling project scheduling problems effectively.** 2

Key words: Multiple Resources Leveling; Symbiotic Organisms Search; Optimization; Construction Management. 1. Introduction The resource management is one of the key success factors for the construction contractors to remain competitive in today's construction business environment. An excellent resource management helps ensure the operational expenses of the project within budget and schedule on time. Construction resources consist primarily of manpower, equipment, materials, funds, and expertise. Obviously, the proper management of these resources plays a significant role to the successful accomplishment of any project. One of the most common problems that often faced by construction managers is the scarcity of resources in the project. The timing of the need of the resources should be determined within project schedules. However, the project schedules generated using network scheduling techniques, such as PERT and CPM, often cause resource fluctuations that are impractical, inefficient, and costly to implement (Martinez and Ioannou 1993). In fact, resource fluctuations become troublesome issue for contractors because hiring and firing the workers necessary to harmonize with fluctuating resource profiles is impractical (Christodoulou, Ellinas and Michaelidou-Kamenou 2010). Thus, contractors are inevitably burdened by a certain percentage of idle resources during periods of low demand, which detracts from project profits. Therefore, resources must be managed efficiently in order to maximize resource expenditures and meet contracted schedules. The process of smoothing out resources, known as resource leveling, has been studied extensively (Doulabi, Seifi and Shariat 2011; Savin, Alkass and Fazio 1996; Son and Skibniewski 1999). Resource leveling attempts to minimize both the demand peak and the fluctuations in the pattern of resource usage (Yan et al. 2005) by optimizing noncritical activities within their available floats while keeping the project duration unchanged. Research on resource leveling has focused mainly on three aspects: (1) single-resource leveling in single-project scheduling (Damci and Polat 2014), (2) multiple-resource leveling in single- project scheduling (Ponz-Tienda et al. 2013), and (3) single-resource leveling in multiple- project scheduling. However, multiple-resource leveling in multiple-project scheduling (MRLMP) is the most typical scenario in the construction and manufacturing industries, a situation that is relatively more complex and difficult to solve and that lacks a standard handling procedure (Guo, Li and Ye 2009). Thus, developing a more efficient optimization algorithm for MRLMP problems and to attain better resource-leveling-problem solutions are essential to improving the management of construction project resources. As optimization problems have varied extensively, a great number of studies have been devoted on the development of new metaheuristic algorithms (Cheng, Tran and Wu 2014). The metaheuristic algorithms perform better than most traditional mathematical techniques in solving modern optimization problems since they do not require substantial gradient information. A

very promising recent development in the field of metaheuristic algorithms is **the** 1

Symbiotic Organisms Search (SOS) algorithm (Cheng and Prayogo 2014). The

SOS algorithm is based on the interactive behavior among organisms in nature. 1

Preliminary studies indicate that the new SOS algorithm is superior over the widely used

Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Differential Evolution (DE), 3

and Bees Algorithm (BA) in solving a various continuous benchmark function and engineering problems (Cheng and Prayogo 2014). Since the SOS algorithm is relatively new, the capability of the SOS algorithm in finding the global solution is very interesting to be further explored and investigated. The aim of this paper is to propose a new discrete optimization model for solving MRLMP problems based on the SOS metaheuristic algorithm (DSOS-MRLMP). In this paper, a methodology to transform the continuous SOS into a discrete one is further provided in this paper because SOS is first suited for continuous optimization problems and MRLMP is considered as a discrete problem. Research contributions This study presents DSOS, a novel optimization algorithm. This algorithm delivers two important contributions: First, it is a first discrete version of the basic SOS algorithm. DSOS transforms continuous-based solutions into discrete-based solutions to fit the MRLMP. Second, the DSOS is more effective and efficient than current widely used evolutionary algorithms, as demonstrated in a numerical construction case study. DSOS outperformed the

genetic algorithm (GA), the particle swarm optimization (PSO), and the differential evolution (DE) 3

in terms of accuracy, stable solutions, and degree of satisfaction. In the remaining parts of this paper, we first review briefly literature related to the establishment of the new optimization model. In the next section, the detailed descriptions of the proposed optimization model for the resource-leveling problem are presented in details. Subsequently, the performance of the newly developed model is demonstrated using numerical experiment and result comparisons. The final section presents conclusions and suggestions directions for future work. 2. Literature Review 2.1 Background of the TCET problem A total of n projects must be started simultaneously in an enterprise. Each project includes multiple activities and each activity uses p resources. Symbols used in related formulas include: the set of activities in the project k is {(ik, jk)} = {Ak, …, Zk}; Rm(t) is the demand for resource m by all n projects on day t; Rmt(ik, jk) is the demand for resource m by activity (ik, jk) on day t; Rm(ik, jk) is the demand for resource m by activity (ik, jk) on one day. TE(ik, jk), TL(ik, jk), Ts(ik, jk), Tf(ik, jk), T(ik, jk), S(ik, jk) represent early start time, late start time, actual start time, actual finish time, duration, and slack time of (ik, jk), respectively. The precedence set of activity (ik, jk) is {(psetk, ik)}. Multiple-resource leveling in multiple-project scheduling differs from conventional 98 resource leveling techniques primarily as follows (Guo, Li and Ye 2009): 99 Firstly, due to differing levels of resource demand, assimilation must transform absolute 100 demand into relative demand in order to enable all the p resources to be comparable in terms 101 of quantity. The relative demand of resource m in all n projects on day t may be expressed as: SRm(t) ? ?Rm(t) / Rmaxm (1) 102 where Rmaxm = max {Rm(t)} denotes the maximum demand for resource m in total n projects 103 on one day and λ is an amplifying coefficient within [1,100] used to increase simulation 104 accuracy. The above formula limits the relative demand for each resource in a total of n 105 projects on every single day to between 0 and λ. 106 Secondly, the weight score wm measures degree of importance for each resource. This 107 paper uses the analytical hierarchy process (AHP) to set the weights of different resources. 108 Larger weight scores correlate to greater priority. 109 The mathematical formulation objective function for multiple-resource leveling in 110 multiple-projects scheduling is: 111 Min RI ? T1 ?tT?1?m?1??wm(SRm(t) ? SRm)2?? p subject to: TE (ik , jk) ? Ts (ik , jk) ? TL (ik , jk) max{Ts (vagk , ik) ? Ts (vagk , ik)} ? Ts (ik , jk) ? TL (ik , jk) Rm(t) ? ?k?1i?k,jkRm(ik, jk); SRm ? T1 ?tT?1SRm(t) n Rm(ik , jk) ? ?????0Rmif(ik:, tjk?) Tifs(:ik ,Tjsk(i)k ,ojrkt)??Ttf?(iTk,f (jikk), jk) S(ik, jk)?TL(ik, jk)?TE(ik, jk) (2) (3) (4) (5) (6) (7) where T is equal to the difference between the maximum of the latest finish time and the minimum of the earliest start time for all n projects. 2.2 Symbiotic Organisms Search algorithm The SOS algorithm is a new metaheuristic algorithm developed by Cheng and Prayogo (Cheng and Prayogo 2014). It is inspired by the biological dependency-based interaction seen among organisms in nature. The dependency-based interaction is often known as symbiosis. Like most population-based metaheuristic algorithms, SOS shares the similar following features: it uses a population of organisms which contains candidate solutions to seek the global solution over the search space; it has special operators that employ the candidate solutions to guide the searching process; it uses a selection mechanism to preserve the better solutions; it requires a proper setting of common control parameters such as population size and maximum number of evaluations. However, unlike most metaheuristic algorithms which have additional control parameters (i.e. GA has crossover and mutation rate; PSO has inertia weight, cognitive factor, and social factor), SOS requires no algorithm-specific parameters. This is considered as an advantage over competing algorithms since SOS does not need additional work for tune the parameters. Improper tuning related to the algorithm-specific parameters might increase the computational time and produce the local optima solution. In the early stage, a random ecosystem (population) matrix is created, each row representing a candidate solution to the corresponding problem. The number of organisms in the ecosystem, so-called the ecosystem size, is pre-determined by the users. The rows in the matrix are called organisms, same as individuals in other metaheuristic algorithms. Each virtual

organism represents a **candidate solution to the corresponding problem** **3**

/ objective. The search begins after the initial ecosystem generated. During the searching process, each organism gains benefit from continuously interacting with one another through three different ways: 1. Mutualism Phase: The phase where one organism is developing a relationship that benefits itself and also the other. The interaction between bees and flowers is a classic example to explain the philosophy of mutualism. 2. Commensalism Phase: The phase where one organism is developing a relationship that benefits itself while does not impact the other. An example of commensalism is the relationship between remora fish and sharks. 3. Parasitism Phase: The phase where one organism is developing a relationship that benefits itself but harms the other. An example of parasitism is

the plasmodium parasite, which **uses its relationship with the anopheles mosquito to pass between human hosts.** 1

These three phases are adopted from the most common symbioses used by organisms to increase their fitness and survival advantage over the long term. During the interaction, the one who receive a benefit will evolve to a fitter organism while the one who is harmed will perish. The mechanisms for updating the best organism will be conducted after one organism has completed their three phases. The phase will operate until the stopping criterion is achieved. The following pseudo-code further summarizes the basic step SOS optimization procedure: 1: Initialization (initial ecosystem, set ecosystem size and maximum iteration) 2: For counter=1 to maximum iteration 3: For each organism in the ecosystem 4: Mutualism Phase 5: Commensalism Phase 6: Parasitism Phase 7: End For 162 8: End For 163 3. Discrete Symbiotic Organisms Search for Multiple Resources Leveling in Multiple 164 Projects (DSOS-MRLMP) 165 This section describes the newly proposed DSOS-MRLMP model in detail. The SOS 166 algorithm plays an important role as the core optimizer in the DSOS-MRLMP model. Fig. 1 167 the overall operational architecture of the proposed algorithm. The objective of the DSOS- 168 MRLMP is to minimize daily fluctuations in resource utilization without changing total 169 project duration. 170 <Insert Fig. 1 here> 171 3.1 Initialization 172 Inputs required by the DSOS-MRLMP optimization model include activity relationship, 173 activity duration, and resource demand. In addition, the user must provide search engine 174 parameter settings such as maximum number of search iterations Gmax and ecosystem size 175 (ecosize). The scheduling procedure uses these inputs in the calculation process to obtain the 176 project duration and resource amount required for each activity. With all the necessary 177 information provided, the model is capable of operating automatically without any human 178 intervention. 179 Prior to beginning the search process, a uniform random generator generates initial 180 ecosystem consists of organisms (feasible solutions): Ecosystem = ? ? ? ? ? = ? ? ??Xecosize?? ? X1 ? ? X 2 ? ? ? X i ? ? X1,1 ? 2,1 ? X ? ? ? X i,1 ? ? ??Xecosize,1 X1,2 X 2,2 X i,2 X ecosize,2 ? ? ? ? ? ? ? Xecosize,D?? ? X1,D X 2,D X i,D (8) A solution for the resource-leveling problem is represented as a vector with D elements as follows: X ? [Xi,1, Xi,2,..., Xi,j,..., Xi,D] (9) where D is the number of decision variables in the problem at hand. D is also the number of non-critical activities in the project network. The index i denotes the ith member in the ecosystem. The vector X represents the start time of D non-critical activities in the network. Because the original SOS operates with real-value variables, a function is employed to convert the start times of those activities from real values to integer values that are constrained within the feasible domain. Xi, j ? Round(LB(j) ? rand[0,1]? (UB(j) ? LB(j))) (10) where Xi, j is the start time of activity j at the individual ith . rand [0,1] denotes a uniformly distributed random number between 0 and 1. LB(j) and UB(j) are the early start time and late start time for activity j. In

multiple resources leveling in the multiple projects scheduling problem, 2

two constraint conditions limit the actual start time of all activities: (1) actual start time must be between the early and late start times and (2) actual start time is limited by the actual start time of its predecessor activities. The first constraint is simple to handle because limits are fixed prior to calculation. However, the minimum limit of the second constraint is unknown prior to calculation and thus more difficult to elicit. For the decision variables of SOS on each dimension is determined in turn, when calculating the actual start time of one activity, actual start time of all activities in its predecessor set Ts(vagk, ik) have been computed, the max{Ts(vagk, ik)+ T(vagk, ik)} has been confirmed simultaneously. 3.2 Mutualism Phase Let Xi be the organism matched to the i-th row of the ecosystem matrix. The organism 3

Xi selects organism Xj as its partner randomly from the ecosystem. Organism Xi is associated to the j-th row of the ecosystem where j is not equal to i.

The mutualistic symbiosis between organism Xi and Xj is modeled in Eqs. (11) and 1

(12). Xinew ? Xi ?rand(0,1)*(Xbest ?Mutual_Vector*BF1) (11) Xjnew ? Xj ?rand(0,1)*(Xbest ? Mutual_Vector*BF2) (12) Mutual_Vector? Xi ? Xj 2 (13) Some notes on the mutualism mathematical model: 1.

rand(0,1) in Eqs. (11) and (12) is a vector of random numbers 3

between 0 and 1. 2. “Mutual_Vector” represents the mutual connection between organism Xi and Xj. 3. Xbest can be understood in two different ways. First, Xbest represents the best organisms in an ecosystem. Second, Xbest can be stated as the current highest state of adaptation to the ecosystem. In this model, the use of Xbest is based on the latter case. 4. Organism Xi might receive a huge benefit when interacting with organism Xj. Meanwhile, organism Xj might only get not so significant benefit when interacting with organism Xi. Here, Benefit Factors (BF1) and (BF2) are determined stochastically as either 1 or 2. This illustrates whether an organism partially or fully benefits from the interaction. 5. Organisms are evolving to a fitter version

only if their new fitness is better than their pre-interaction fitness. 3

In this case, the old Xi and Xj will be replaced by Xi new and Xj new. This type of mechanism is similar to greedy selection techniques. 6. For each organism Xi, this interaction counts for two function evaluations. 3.3 Commensalism Phase After the mutualism phase is finished, the organism Xi selects again a new partner randomly from the ecosystem, organism Xj.

In this circumstance, organism Xi attempts to benefit from the interaction but organism Xj neither benefits nor suffers from the relationship. The 3

commensal symbiosis between organism Xi and Xj is modeled in Eq. 3

(14). Xi new ? Xi ? rand(?1,1) *(Xbest ? X j) (14) Some notes on the commensalism mathematical model:

1. rand(-1,1) in Eq. (14) is a vector of random numbers between -1 and 1. 2. Xbest reflects the current highest state of adaptation to the ecosystem, similar to those in the mutualism phase. 3.

Organism Xi is updated by Xi new **only if its new fitness is better than its pre-interaction fitness. 4.** 1

For each organism Xi, this interaction counts for one function evaluation. 3.4 Parasitism Phase After the commensalism phase is completed, the organism Xi selects again a new organism randomly from the ecosystem, organism Xj. In parasitism,

organism Xi is given a role similar to the anopheles mosquito through the creation of an artificial parasite called 1

“Parasite_Vector”. Organism Xj serves as a host to the Parasite_Vector. During the interaction, the Parasite_Vector tries to kill the host Xj and replace Xj in the ecosystem. The organism Xi will gain a benefit because his cloning will be exist in the ecosystem, and Xj will have to suffer and die. The creation of Parasite_Vector is described as follows: 1. Initial

Parasite_Vector is created in the search space by duplicating organism Xi. 1

Some decision variables from the initial Parasite_Vector will be modified randomly in order to differentiate Parasite_Vector with organism Xi. 2. A random number is created within a range from one to the number of decision variables. This random number represents the total number of modified variables. 3. The location of the modified variables is determined stochastically. 11 4. Finally, the variables are modified using a uniform distribution within the range of the search space. The Parasite_Vector is ready for the parasitism phase. Both Parasite_Vector and organism Xj

are then evaluated to measure their fitness. If Parasite_Vector has a better fitness value, it will kill organism Xj and assume its position in the ecosystem. If the fitness value of Xj is better, Xj will have immunity from the parasite and the Parasite_Vector will no longer be able to live in that ecosystem. 1

For each organism Xi, this interaction counts for one function evaluation. 3.5 Stopping Condition The optimization process terminates when a user-set stopping criterion is met. This stopping criterion is often set as the maximum iteration Gmax or the maximum number of function evaluations (NFE). Search process termination identifies the optimal solution. The project schedule and its corresponding resource histogram may then be constructed based on the optimal start time for activities. 4. Case Study A case study adapted from Guo, Li and Ye (2009) was used to demonstrate the capability of the newly developed SOS-MRLMP model. In this case study, an enterprise must start two projects with same total project duration. Fig. 2 shows the precedence relationships of the network projects. Each activity in both projects uses three resources (R1 human, R2 fund, R3 equipment) and has a certain duration D that is indicated above the arrow line. <Insert Fig. 2 here> Based on the importance of each resource, the AHP method makes pairwise comparison of each resource. The comparison matrix is obtained as follows: R1 ? 1 3 5? R2 ?? 3? 1 1 3 ? R 3 ??5?1 3?1 1?? ? Consistency inspection demonstrates that this comparison matrix is acceptable. Weights 271 for each resource are set as: w1 ? 0.637, w2 ? 0.258, w3 ? 0.105 . Consequently, the objective 272 function for the case study is calculated as follows: 273 min RI ? 118 t1??81?? 0.637(SR1(t) ? SR1(t))2 ? 0.258(SR2(t) ? SR2(t))2 ? 0.105(SR3(t) ? SR3(t))2?? ? ? 0 ? Ts (A1) ? 7 0 ? Ts (I1) ? 15 ? 0 ? Ts (B1) ? 3 0 ? Ts (A2) ? 9 274 s.t ? ? Ts (B1) ? 5 ? Ts (C1) ? 8 0 ? Ts (C2) ? 15 ? 0 ? Ts (F1) ? 6 5 ? Ts (D2) ? 9 ?Ts (B1) ? 4 ? Ts (G1) ? 10 ? 5 ? Ts (G 2) ? 7 ? 0 ? Ts (H1) ? 3 5 ? Ts (H 2) ? 13 4.1 Optimization Result for the DSOS-MRLMP Application of the DSOS-MRLMP model reduces significant fluctuations in resource use. This study set parameters for the DSOS optimizer based on proposed values from the literature and several experiments as shown in Table 1 (Cheng and Prayogo 2014). Fig. 3 shows the network resource profile for the project at initialization and after being leveling by DSOS-MRLMP optimization. <Insert Table 1 here><Insert Fig. 3 here> 4.2 Result Comparisons Three different algorithms were used to verify the comparative performance of the proposed model (DSOS-MRLMP). These algorithms were: DE (Storn and Price 1997), PSO (Clerc 2006), and GA (Haupt and Haupt 2004). For comparison purposes, all four algorithms used an equal number of function evaluations, had population sizes of 100, and used a maximum of 200 generations. In GA, the constant mutant and crossover probability factors were set at 0.5 and 0.9, respectively. In PSO, the two learning factors, c1, c2, were both chosen at 2.05, and the inertia factor w is set in the range of 0.3–0.7. DE control parameters is set as 13 0.5 and 0.8 for mutant factor F and crossover probability Cr, respectively. Twenty-five independent runs were carried out for all experiments. Table 2 lists the optimal results, optimal non-critical-activity start times obtained from the new proposed model, and other benchmark algorithms. RIm in Table 2 is the resource intensity for single resource m: min RI ? 118 t?1?81?Rm(t) ? Rm(t)? , Rm ? 118 2 T ? Rm (t) t ?1 <Insert Table 2 here> As shown in Table 2, the optimal resource intensity (RI) obtained by DSOS was, respectively, 94.9%, 2.1%, 6.9%, 8.6% less than the initial schedule, DE, PSO, and GA. Fig. 4 presents the resource profile after being optimized by each algorithm. <Insert Fig. 4 here> To evaluate the stability and accuracy of each algorithm, optimization performance was expressed in terms of best result found (best), average result (avg), standard deviation (std), and worst result (worst) after 25 runs (Table 3). The best and worst results demonstrate the capacity of each algorithm to find the optimal solution for all of the performance measurement metrics. Average and standard deviation are two additional characteristics that describe solution quality. Standard deviation occurs in cases when algorithms are not able to generate optimal solutions in all executions. <Insert Table 3 here><Insert Fig. 5 here> As shown in Table 3, the performance of the DSOS-MRLMP is competitive in terms of accuracy and stability. It is clearly shown that the DSOS-MRLMP is only able to find optimal solutions in fitness function. Further, in terms of average results, DSOS-MRLMP performed the best of the considered algorithms because it generated the lowest average fitness solution, with a value of 4.740 and deviation value of 0.294. Fig. 5 illustrates the best fitness value results of different approaches by number of iterations. 4.3 Statistically test A hypothesis test was performed to further demonstrate the superiority of the DSOS performance over that of benchmark algorithms. Because all indicator comparisons demonstrated that the DE performed better on average than either PSO or GA, the hypothesis tests only considered DSOS and DE. A one-tailed t-test with equal sample sizes and unequal and unknown variances analyzed the following hypothesis tests: Hypothesis: DSOS versus standard DE in term of resource intensity (RI) (Table 3) H0: There is no difference in RI of the DSOS algorithm and that of the standard DE algorithm. H1: The DSOS algorithm is significantly better than the standard DE algorithm. DSOS: s1 ? 0.294; DE: s2 ? 0.424; n1 ? n2 ? n ? 25; v? ? s?12s1/2 n/1n?12 ?? s?22s/22 n/2n?22?2 ? ?0?.02.924924/2 2/52?52 ??0?.04.244224/2 2/52?52?2 ? 42.7 (closest to 43) n1 ? 1 n2 ? 1 25 ? 1 25 ? 1 Critical value: with significant level of t-test ? ? 0.05 ; ? ? 43 ; we have t? ;? ? t0.05;18 ? 1.681 Statistical test: t ? ? x1 ? x2 ? s12 / n1 ? s22 / n2 0.3002 / 25 ? 0.4242 / 25 ? ?4.740 ? 5.400? ? ? 6.396 ? ?1.681 ? ?t0.05;43 where n is the sample size (number of experimental runs), ? is the degrees of freedom used in the test, s12 and s22 are the unbiased estimators of the variances of the two samples (DSOS and DE). The denominator of t is the standard error of the difference between two means x1 , x2 (average). The statistical test value above is smaller than critical value. Therefore, H0 is rejected. The proposed DSOS is thus demonstrated to be statistically superior to the standard DE in terms of resource intensity. 5. Conclusions This paper uses DSOS to solve the problem of multiple-resource leveling in the context of multiple-projects scheduling (MRLMP). An application example is analyzed to illustrate the effectiveness of the proposed model and to demonstrate the capabilities of the model in generating an optimal schedule that eliminates undesirable resource fluctuations and resource idle times. Obtained results indicate excellent performance of the SOS algorithm in solving the MRLMP. The search strategy using organism interactions was important to the extraordinary performance of DSOS algorithm. The proposed DSOS strategy differs significantly from previous metaheuristic algorithms such as GA, PSO, and DE because it simulates natural patterns using the three strategies of mutualism, commensalism, and parasitism of the SOS algorithm to gradually improve candidate solutions. These

three phases of the SOS **algorithm are simple and** **3**

require only few additional lines of code on MATLAB platforms. Another major advantage of SOS over competing algorithms is the small number of parameters that must be tuned. The SOS algorithm has broad application potential because the model is easily modifiable for solving many other classes of single-objective optimization problems in the construction management field such as resource-allocation and resource-constrained problems. Moreover, resource-leveling problems in the realm of total-project-cost minimization are frequently encountered in construction management. Tradeoffs between time and cost are necessary to improve overall construction project benefits. Further work is necessary to address these issues in order to apply DSOS to the resolution of complicated resource-leveling problems 359 that consider multi-objective optimizations. Extending the current DSOS from a single- 360 objective to a multi-objective format is an interesting direction for further research. 361 Table 1 Settings for DSOS-MRLMP parameters Input parameters Notation Setting Number of decision variables D 12 Population size NP 100 Amplification coefficient ? 30 Maximum generation Gmax 200 Table 2 Comparison of optimal performance for algorithms Item RI RI1 RI2 RI3 Actual start time of non-critical activities A1 B1 C1 F1 G1 H1 I1 A2 C2 D2 G2 H2 Initial 89.46 76.95 1169 123.8 GA 5.327 1.06 13.76 9.17 PSO 4.897 0.84 26.65 7.95 DE 4.652 0.84 21.97 5.73 DSOS 4.558 0.62 21.31 9.51 0 0 5 0 4 0 0 0 0 5 5 0 3 8 0 9 0 15 8 12 9 6 0 0 8 6 10 3 12 0 15 8 5 0 3 9 0 10 0 12 8 15 9 6 3 0 8 0 8 0 12 8 15 9 5 5 13 13 13 13 Note that RI is resource intensity. Table 3 Comparison of results for the DSOS-MRL and benchmarked algorithms Performance Measurement GA PSO DE SOS Fitness value Best Avg. Std. Worst 5.327 6.907 2.112 13.385 4.897 6.327 0.742 7.968 4.652 5.400 0.424 6.237 4.558 4.740 0.294 5.339 Start Optimizer s parameter setting DSOS Optimizer Mutualism Phase Project information Initialization Commensalism Phase Scheduling module Parasitism Phase Ecosystem Searching Termination [X1, 1, X1, 2, , X1, D] ... Yes [Xecosize, 1, , Xecosize, D] Optimal Activity Start-Time [X1, X2, , XD] No Stop Fig 1 Flowchart for the DSOS-MRLMP A1(2/8/4/6) 2 B1(2/7/2/5) 3 1 D1(4/20/4/8) F1(3/7/3/4) 6 G1(1/4/2/3) C1(1/6/3/5) 4 5 E1(2/7/2/5) 7 10 (R1/R2/R3/D) J1(3/15/2/5) 11 H1(2/14/5/10) 8 I1(4/18/3/3) 9 Time 0 1 2 3 4 5 6 7 8 9 10 11 12 a) Network of Project 1 13

14 15 16 17 18 A2(4/10/4/4) 2 D2(3/7/3/4) 3 1 B2(2/6/3/5) 4 E2(2/5/3/5) 5 F2(4/13/5/3) G2(4/12/3/3) 7 C2(5/20/5/3) 8 H2(4/16/4/5) 9 6 (R1/R2/R3/D) I2(2/8/3/5) 10 Time 0 1 2 3 4 5 6 7 8 9 10 11 12 b) Network of Project 2 13 14 15 16 17 18 Fig 2 Networks of two projects a) Resource demand of initial network b) Resource demand after being DSOS optimizer 120 120 Resource 1 Resource 1 100 Resource 2 100 Resource 2 Resource 3 Resource 3 80 80 Resources 60 Resources 60 40 40 20 20 0 0 0 2 4 6 8 10 12 14 16 18 0 2 4 6 8 10 12 14 16 18 Days Days Fig 3 Resource profile of projects a) Resource demand optimized by DSOS optimizer b) Resource demand optimized by DE optimizer 60 60 Resources 40 Resource 1 Resource 2 Resource 3 20 Resources 40 Resource 1 Resource 2 Resource 3 20 0 0 0 2 4 6 8 10 12 14 16 18 0 2 4 6 8 10 12 14 16 18 Days Days c) Resource demand optimized by PSO optimizer d) Resource demand optimized by GA optimizer 60 60 Resources 40 Resource 1 Resource 2 Resource 3 20 Resources 40 Resource 1 Resource 2 Resource 3 20 0 0 0 2 4 6 8 10 12 14 16 18 0 2 4 6 8 10 12 14 16 18 Days Days Fig 4 Resource profile of projects by different algorithms 30 GA PSO 25 DE DSOS Project Resource Intensity 20 15 10 5 0 20 40 60 80 100 120 140 160 180 200 Iterations Fig 5 Best project resource intensity curves for algorithms 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 97 161 270 1 2 3 4 5 6 7 8 9 10 12 14 15 16 17 1 2 3 4 5 6 7 8