文章编号　1672-6634(2019)04-0101-10　　**DOI**　10.19728/j.issn1672-6634.2019.04.015

装配式建筑配送中带时间窗 VRP 问题研究

李俊青[1,2]　宋美娴[1]　邓佳文[1]　韩云琦[2]

(1.聊城大学 计算机学院,山东 聊城 252059;2.山东师范大学 信息科学与工程学院,山东 济南 250014)

摘　要　近年来,带时间窗的车辆路径问题(vehicle routing problem with time window,VRPTW)得到了广泛关注和研究.装配式建筑是近年来发展的一种新型建筑类型,预制构件配送过程中会带来诸多复杂工程问题.本文以装配式建筑配送为研究背景,分析了当前 VRPTW 相关文献的研究现状,建立基于 VRPTW 的扩展模型,并采用智能优化算法进行求解.以经典的 SOLOMN 算例作为扩展,随机生成 18 个不同结构的算例进行测试,实验结果验证了本文所提出算法的有效性.

关键词　车辆路径规划问题;装配式建筑配送;智能优化算法;预制构件;时间窗

中图分类号　F426.92;TP18　　**文献标识码**　A

0　引言

装配式建筑是近年来出现的一种建筑新形势,大量的建筑构件由车间生产加工完成,通过车辆运送到施工现场.由于采用现场装配作业方式,相比传统的现浇作业方式大大减少了空气污染等对环境的影响,是国家十三五发展规划重点提及的技术之一.在装配式建筑中,包含的预制构件种类主要有外墙板,内墙板,叠合板,阳台,空调板,楼梯,预制梁,预制柱等.上述预制构件的配送以及路径选择问题是一类复杂的工程问题.该类问题可以看做车辆路径问题(vehicle routing problem,VRP)的一类扩展问题.

VRP 是由 Dantzig 和 Ramser 于 1959 年提出的[1].目前,关于 VRP 的相关研究主要集中于带时间窗的 VRP(vehicle routing problem with time window,VRPTW).对 VRPTW 的研究文献分析,按照约束的不同进行分类,主要包括开放式、带回程式、同时访问式、不确定时间式、带充电站式等[2,3].开放式 VRPTW 是指车辆配送货物后不必回到仓库,这种情况主要出现在第三方物流中.研究该类问题的文献主要有迭代局部搜索算法(iterated local search,ILS)[4]、禁忌搜索算法(tabu search algorithm,TSA)[5].带回程式 VRPTW 是指车辆配送货物的同时,需要回收客户点的物品,即同时取送货.研究该类问题的文献主要包括自适应大规模邻域搜索算法(adaptive large neighbourhood search)[6],两阶段元启发式(meta-heuristic)算法[7]等.同时访问式 VRPTW 是比较新颖的一种车辆路径问题,该类问题要求一个客户点必须由多辆车同时服务,其中,文献[8]为该类问题建立了混合整数规划模型,并采用自适应邻域搜索算法求解,文献[9]则采用约束规划和邻域搜索相结合的策略.不确定时间式 VRPTW 综合考虑各类不确定因素带来的影响,文献[10]研究了派送时间和服务时间随机的环境下,采用多目标局部迭代搜索算法最小化运营成本并最大化服务水平,文献[11]为该类问题设计了一种变邻域搜索(variable neigh-borhood search,VNS)和模拟退火(simulated annealing.SA)相结合的算法,文献[12]则考虑在家庭保健治疗配送中,将病人所需药品量视作模糊量,

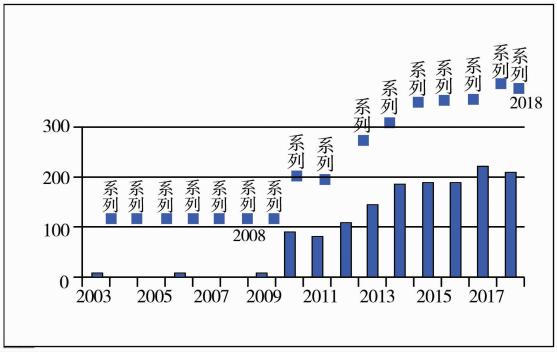


图 1　近年来 VRPTW 文献对比分析图

收稿日期:2018-11-20

基金项目:国家自然科学基金项目(61773192)资助

通讯作者:李俊青,男,汉族,博士,教授,博士生导师,研究方向:人工智能,智能优化,E-mail:lijunqing@lcu-cs.com.

并采用遗传算法和随机仿真相结合的算法进行求解.图1展示了近年来VRPTW相关研究论文发表情况.

关于装配式预制构件的配送的文献还很少,刘宏令以京津冀地区为研究对象,对上述地区的装配式配送问题进行深入分析,通过增加配送中心数目的方式解决装配式配送成本过高的问题,并对配送中心的选址进行优化[13];李萍萍通过对不同种类的装配式预制件赋予不同的质量,对不同客户点设置不同的需求量,建立装配式配送问题模型,并运用人工鱼群算法对问题进行求解[14];彭星在装配式预制件配送过程中根据客户需求量和物流成本等因素随时间变化的动态特征,对配送点进行动态调整,以总成本最小化为目标建立动态选址问题模型,并通过遗传算法对模型进行求解、验证[15].

综合分析上述文献可见,当前VRPTW问题已经得到了广泛研究,装配式预制构件配送过程可以以VRPTW为基础建立模型并进行求解,但目前还缺乏相关文献,亟待设计算法进行求解.人工蜂群(artificial bee colony,ABC)算法是一种新型群体智能优化算法[16],目前已在连续函数优化、调度优化等多个领域得到了广泛应用并证明了其有效性[17-22].本文采用一种改进的ABC算法,针对装配式预制构件配送问题,以VRPTW为基础建立模型并进行求解.

1 VRPTW问题描述

1.1 VRPTW描述

经典的VRP问题是旅行商问题(travelling salesman problem,TSP)的一个扩展,二者的主要区别体现在,VRP问题研究一类有多辆车配送货物到多个客户点的路径优化问题,而TSP问题研究一辆车配送物品到多个城市的问题.VRP和TSP的共同约束是,每个客户点只能配送一次,车辆容量有约束限制,即车辆运送货物不能超过该车辆最大容量.带时间窗的VRP(vehicle routing problem with time window, VRPTW)是经典VRP的一个典型扩展,也是现实物流配送中的典型应用.VRPTW增加了客户点服务时间窗的约束,即每个客户点定义了各自服务时间窗,VRPTW又可以分为硬时间窗和软时间窗.在硬时间窗VRPTW中,早于服务时间窗到达的车辆需要等待,晚于服务时间窗的车辆则不能为该客户点服务.在软时间窗VRPTW中,提前或延后到达的车辆可以继续为客户点服务,但系统要增加惩罚成本.

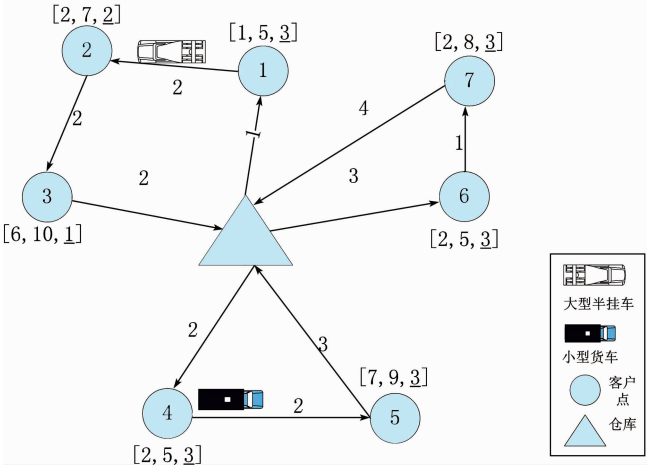


图2 VRPTW示例图

图2给出了一个VRPTW示例图,图中分别表示了四种图形,即两种运送车辆、客户点和仓库.图中每个客户点用圆形表示,客户点附近的三个数字表示出了该客户点的服务时间窗和服务时间.譬如,如果时间单位为分钟,则客户点1的服务时间窗是[1,5],即如果服务车辆在1到5 min之间到达客户点1,则服务满意度为100%,服务时间为3 min.从仓库到达客户点1的距离为1min的车程,即车辆目前到达客户点1的时间为1,正好处在其服务时间窗内.在客户点1服务3 min后,该车辆在时刻4开始经由客户点1开往客户点2;到达客户点2的时刻为6,也处于客户点2的时间窗内,车辆离开客户点2的时刻为8,以此类推,最后回到仓库的时刻为13.

1.2 VRPTW建模

VRPTW建模所需参数和符号下标n:客户点数量,m:当前解派送车辆的数量,Q:车辆负载最大限制,K:派送车辆的数量限制,L:每辆车最大工作时间,d_i:客户点i的派送货物需求量,t_{ij}:客户点i到客户点j之间的路径长度,a_i:客户点i的最早可能服务时间,即服务时间窗的最小值,b_i:客户点i的最晚可能服务时间,即服务时间窗的最大值,p:如果车辆早于客户点的最早可能服务时间到达,对于软时间窗VRP,给予的惩罚成本,h:如果车辆晚于客户点的最晚可能服务时间到达,给予的惩罚成本,η:服务时间早于其最早可能服务时间的客户点数量,μ:服务时间晚于其最晚可能服务时间的客户点数量,β:客户满意度,即服务时间处

于客户点服务时间窗内的比例，γ：服务时间处于客户点服务时间窗内的客户数量的最低限制，t_i：车辆到达客户点 i 的时刻，s_i：车辆在客户点 i 的服务时间长度，y_{ij}^k：0-1 决策变量，如果车辆经由客户点 i 到客户点 j，则变量设置为 1；否则为 0.

$$\min K, \tag{1}$$

$$\min Z=\sum_{i=0}^{N}\sum_{j=1}^{N}\sum_{k=1}^{K}(t_{ij}\times y_{ij}^k)+p\times\sum_{i=1}^{N}\max(a_i-t_i,0)+h\times\sum_{i=1}^{N}\max(t_i-b_i,0), \tag{2}$$

$$\text{s.t. }\sum_{i=0}^{N}\sum_{j=1}^{N}[(t_{ij}+s_i)\cdot y_{ij}^k]\leqslant L,k=1,2,\cdots,K, \tag{3}$$

$$\sum_{i=0}^{N}\sum_{j=1}^{N}(d_i\times y_{ij}^k)\leqslant Q,k=1,2,\cdots,K, \tag{4}$$

$$\sum_{i=0}^{N}\sum_{k=1}^{K}y_{ij}^k=1,j=1,2,\cdots,N, \tag{5}$$

$$\sum_{i=0}^{N}y_{ie}^k=\sum_{j=0}^{N}y_{ej}^k,e=1,2,\cdots,N;k=1,2,\cdots,K, \tag{6}$$

$$\sum_{k=1}^{K}\sum_{j=1}^{N}y_{0j}^k=K, \tag{7}$$

$$\sum_{k=1}^{K}\sum_{j=1}^{N}y_{i0}^k=K, \tag{8}$$

$$K\geqslant K_{\min}=\lfloor\sum_{i=1}^{N}d_i/Q\rceil+1, \tag{9}$$

$$y_{ij}^k\in\{0,1\},i,j=0,1,\cdots,N;k=1,\cdots,K, \tag{10}$$

$$\beta=1-(\eta+\mu)/N\geqslant\gamma, \tag{11}$$

式(1)是问题的第一个目标，即最小化系统所用的车辆数量，式(2)描述了问题的第二个目标，即最小化总成本，这些成本包括三部分，即车辆派送成本、早于客户点服务时间窗开始服务的惩罚成本、晚于客户点服务时间窗开始服务的惩罚成本. 式(3)给出了第一个约束条件，即车辆 k 的派送总时长不能超过系统最大限制，其中派送总时长包括派送路途耗费时间和客户点的服务时间. 式(4)约束限制车辆 k 的总负载不能超过系统给定的最大负载量. 式(5)中描述了经由客户 i 到客户 j 的车辆只能有一个，即每个客户的后续客户点只能有一个. 式(6-8)限制了每个客户点出入的车辆数量保持一致，且总的车辆数量为 K. 式(9)给出了系统至少需要的车辆数量. 式(10)限定了决策变量的是一个 0-1 变量，式(11)限定了客户满意度的取值范围.

1.3　VRPTW 经典算例描述

经典的 VRPTW 算例，如 SOLOMN 算例，包括 56 个算例，每个算例中包含 100 个客户点，客户点的布局分为三大类，即 17 个聚合程度较高的 C(Clustering)系列算例、23 个聚合程度分散的 R(Random)系列算例和 16 个聚合程度处于中间状态的 RC 系列. 图 2 展示了两种不同场景的客户点分布图. C 系列算例的主要特点包括：(1) 客户点聚合程度较高，多个客户点形成一簇；(2) 客户点的需求量相对较大，因而车辆只能配送有限个客户点；(3) 客户点的时间窗长度相对较大. R 系列算例的主要特点包括：(1) 客户点分散程度较高；(2) 客户点的需求量很小，因而车辆能配送足够多的客户点；(3) 客户点的时间窗长度相对较小. 上述特点决定了不同结构的 VRPTW 问题有不同的问题特性，因而应该采用不同的启发式规则进行求解.

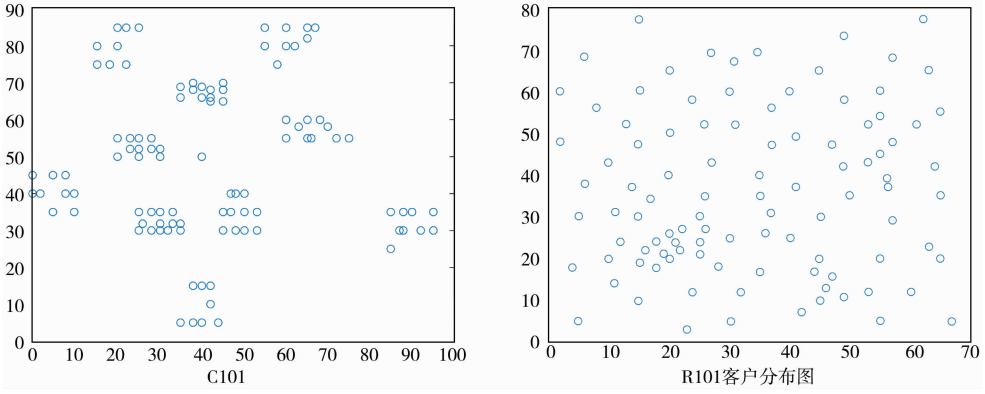


图 3　SOLOMN 经典算例场景图

2 算法设计

2.1 人工蜂群算法

人工蜂群(Artificial Bee Colony,ABC)算法是由 Karaboga 等[16]提出的一种新型群体智能优化算法,是模拟蜜蜂寻找食物的过程而演化的仿生过程.与传统智能优化算法如遗传算法相比,ABC 算法主要由三类蜜蜂协作完成食物源搜索过程,即雇佣蜂(Employed bee)、跟随蜂(Onlooker bee)和侦查蜂(Scout bee).雇佣蜂的主要任务是在分派的食物源完成局部搜索,即为分派的食物源找到更好的解;跟随蜂的主要任务是在蜂巢等待雇佣蜂回到蜂巢,并根据雇佣蜂的搜索结果完成进一步的局部搜索;侦查蜂的功能是在某个食物源在指定迭代次数还不能有更新的情况下,完成进一步的全局搜索.

ABC 算法中基本控制参数包括:解集大小 SN,解无更新而被丢弃的周期大小 Ls,雇佣蜂数目 Es,跟随蜂数目 Os,侦察蜂数目 Ss 和终止条件.ABC 算法的关键过程描述如下[16-22].

(1) 初始解的产生.在基本 ABC 算法的设计中,针对连续优化问题的初始解一般采用随机策略产生.记 $V_i=\{v_i^1,v_i^2,\cdots,v_i^n\}$ 代表第 i 个食物源或解,其中 n 表示问题维度大小,则初始解 i 产生的方法如

$$v_i^j=v_{\min}^j+\mathrm{rand}[0,1](v_{\max}^j-v_{\min}^j), \tag{12}$$

式中 $j=1,\cdots,n;i=1,\cdots,SN$,$v_{\max}^j$ 和 $v_{\min}^j$ 分别表示维度 j 的上限和下限值.

(2) 雇佣蜂策略.雇佣蜂完成局部搜索的过程,假设当前雇佣蜂分配的解 i,则首先雇佣蜂随机选择当前解群体中的一个解,记为 k,产生一个新的邻域解 new 的过程如

$$v_{\mathrm{new}}^j=v_i^j+\mathrm{rand}[-1,1](v_i^j-v_k^j,) \tag{13}$$

式中 v_{new} 表示新产生的解,v_{new}^j 表示新解的第 j 维数值,$v_i^j-v_k^j$ 表示两个解的第 j 维度的差值.

(3) 跟随蜂策略.侦查蜂在等待雇佣蜂回到蜂巢后,根据侦查蜂得到的食物源的状态,采用公式(14)轮盘赌注的方法,选择较好的食物源,即选择概率较大者,继续应用公式(13)做进一步的挖掘搜索.

$$p_i=\frac{f_i}{\sum_{j=1}^{SN}f_j}, \tag{14}$$

式中 f_i 表示解 i 的适应度值,p_i 表示解 i 的选择概率.

(4) 侦查蜂策略.在基本 ABC 算法中,当某个解在迭代 Ls 次还没有任何更新时,采用公式(12)随机产生一个解替换该解.

通过 ABC 算法的基本流程分析可见,ABC 算法通过雇佣蜂完成局部挖掘搜索的过程,通过雇佣蜂完成群体解的协作搜索,通过侦查蜂完成局部搜索.

2.2 问题编码

本文采用二维数组的方式编码一个解,二维数组的第一维表示每一辆车,对于每辆车创建一个数组,包含该车辆服务的客户点序列,客户点序号的先后顺序表示这些客户点的服务次序.图 4 给出了一个 3 辆车、9 个客户点的解,图中第一辆车服务的客户点序列是{2,4,5,1},其中"0"号表示仓库,第二辆车服务的客户点序列是{3,6,7},第三辆车服务的客户点集合是{8,9}.

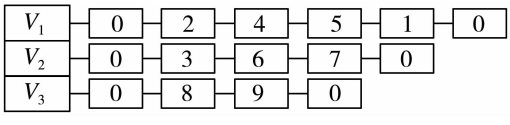


图 4 编码示例图

2.3 问题解码

由问题解码方式可见,编码数组中仅仅给出了每辆车服务的客户点集合,以及客户点服务的顺序关系.在编码中并未给出在每个客户点车辆的到达时刻,因而无法判断车辆是否在客户点的服务时间窗内到达,即对于硬时间窗的 VRPTW 问题,无法保证编码是否可行.为解决非法解问题,本文针对硬时间窗,在解码过程中如果出现非法解,则迭代 R_n 次用于执行 2.4 解的编码修复策略.需要特别指出的是,参数 R_n 表示了解的修复迭代次数,如果 R_n 过小则可能找不到合法解,相反如果 R_n 过大,则可能会造成计算资源的浪费.

2.4 编码修复策略

当某个解在解码过程中出现超出客户点服务时间窗的现象时,对于硬时间窗 VRPTW 问题,则该解为非法解. 函数 repair 给出了修复策略的伪代码,该策略算法时间复杂度为 $O(n^2m)$. 分析可见,编码修复策略并不能保证每次都能转化一个非法解为可行解,当无法添加车辆而又有客户点无法插入到当前任何一个车辆时,该解为不可行解,则放弃该解.

```
Procedure repair()   //修复策略过程.
输入   一个非法解.
输出   一个可行解.
1   for each vehicle i do //循环每一辆车 i.
2       for each customer j assigned on i do //循环车辆 i 上的每个客户点 j.
3           if j violates its time window then //如果 j 超出时间窗.
4               Delete customer j from vehicle i.
5           end.
6       end.
7   end.
8   Store all of the deleted customers into a set DS.
9   for each customer j in DS do //循环每一个删除的客户点.
10      Try to insert customer j into all the current vehicles.
11      if j cannot be inserted then //如果 j 不能插入到当前所有车辆.
12          if number of vehicle has not exceeded then//如果车辆数未超限制.
13              Add a new vehicle and service customer j //添加一辆新车.
14          else
15              Discard the solution and stop the procedure .
16          end.
17      end.
18  end.
```

2.5 初始解生成策略

初始解的产生过程,即为某个车辆安排客户点并排序的过程. SOLOMN 的 PFIH 策略是一种通用的初始解的产生方法,该策略算法时间复杂度为 $O(n^3m)$. 其具体过程描述如

(1) 首先,为当前车辆 i 选择下一个客户点的策略过程,循环所有剩余未调度的客户点,计算每个客户点 j 插入当前车辆所有位置的费用,找到费用最小的位置 p_j,即如果 j 作为下一个安排服务的客户点则其插入位置为 p_j. 客户点 j 插入到车辆 i 的位置 u 的费用计算如

$$c_1(k,u,h)=\alpha_1 c_{11}(k,u,h)+\alpha_2 c_{12}(k,u,h),\alpha_1+\alpha_2=1,\alpha_1,\alpha_2\geqslant 0, \tag{15}$$

$$c_{11}(k,u,h)=t_{ku}+t_{uh}-\mu t_{kh},\mu\geqslant 0, \tag{16}$$

$$c_{12}(k,u,h)=w_{h'}-w_h, \tag{17}$$

式中 k 和 h 表示当前车辆上位置 $u-1$ 和 u 的客户点,$w_{h'}$表示插入客户点 j 到位置 u 后,客户点 h 的开始服务时间,μ 为系统参数.

(2) 计算每个客户点的 c_2 值,找到最小 c_2 值的客户点 j^* ,记为下一个服务的客户点,计算如

$$c_2(k,u,h)=\beta_1 R_d(u)+\beta_2 R_t(u),\beta_1+\beta_2=1,\beta_1\geqslant,\beta_2>0\ , \tag{18}$$

式中,$R_d(u)$ 和 $R_t(u)$ 分别表示在位置 u 插入新的客户点 j^* 后,车辆 i 的总路径费用和部分路径时间代价.

由上述分析可见,在客户点信息和车辆数量一定的情况下,PFIH 策略每次运行都会产生一样的解,单纯采用 PFIH 初始化无法保证解集的多样性. 本文提出了一种改进的 PFIH 策略(Improved PFIH,IPFIH),具体步骤:

Algorithm IPFIH.

1 while k less than $P_{size}-1$ do //P_{size}是初始解集大小.

2 Randomly sequence each customer and store them into a set named SS //对所有客户点随机排序,存入数组 SS 中.

3 for each customer j do //循环每一个客户点 j.

4 Let $mp=0, mv=L$ //mp 表示客户点 j 的最好插入位置,L 是极大值.

5 if customer j can be inserted into vehicle i then.

6 for each position u in the current vehicle i then //循环车辆 i 的所有位置.

7 $c_{11}(k,u,h) = t_{ku} + t_{uh} - \mu t_{kh}, \mu \geqslant 0$.

8 $c_{12}(k,u,h) = w_h - w_h$.

9 $c_1(k,u,h) = \alpha_1 c_{11}(k,u,h) + \alpha_2 c_{12}(k,u,h), \alpha_1 + \alpha_2 = 1, \alpha_1, \alpha_2 \geqslant 0$.

10 if $c_1(k,u,h) < mv$ then.

11 $mv = c_1(k,u,h)$; $mp = u$.

12 end.

13 Insert customer j into the position mp in the current vehicle.

14 end.

15 end.

16 else

17 if number of vehicle has not exceeded then//如果车辆数未超限制.

18 Add a new vehicle and service customer j //添加一辆新车.

19 else

20 Discard the solution and stop the procedure.

21 end.

22 end.

23 end.

24 end.

25 Generate a solution by using the SOLOMN PFIH approach.

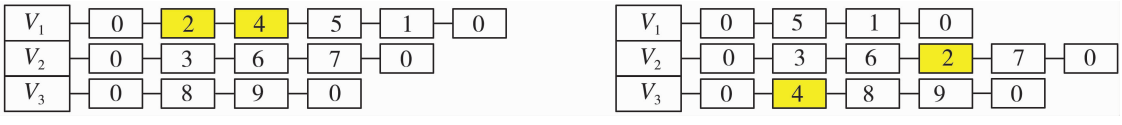
26 Store all the generated solutions into the current population.

分析可见,初始群体为 P_{size},本文给出的策略是,在生成其中 $P_{size}-1$ 解时采用随机策略,算法时间复杂度为 $O(n^2m)$,即随机打乱客户点的次序,因而增加了种群的多样性. 为同时兼顾算法求解性能,选择一个解采用 SOLOMN 的 PFIH 策略.

2.6 局部搜索策略

局部搜索是目前智能优化算法常用的策略之一,本文提出了一种变长度的局部搜索(Variable Length Local Search,VLLS)策略,具体算法描述如:(1) 在当前所有车辆中随机选择一个车辆 i;(2) 在所选车辆中随机选择 S_L 个客户点,其中 S_L 代表了搜索的强度,较大的值代表了较精细的搜索;(3) 循环每一个选择的客户点 j,在其余车辆中为其寻找可插入的最好的位置.

图 5 给出了 VLLS 局部搜索策略的示意图,图 5(a)表示了原解编码,所选车辆为 1 号车辆,$S_L=2$,则在 1 号车辆随机选择两个客户点 2 和 4. 之后为客户点 2 选择最好插入位置是 2 号车辆的第 3 个位置,为客户点 4 选择的最好插入位置是 3 号车辆的第 1 个位置. 局部搜索后新解的示意图见图 5(b).



(a) 原解 (b) 新的邻域解

图 5 邻域搜索示意图

图 6 给出了变长度的局部搜索 S_L 参数的变化图,假设迭代次数最大为 100 次,则随着迭代次数不断增

大，S_L不断增大，表明搜索强度不断加强，因而可以确保在迭代后期精细化搜索，提高算法搜索能力.

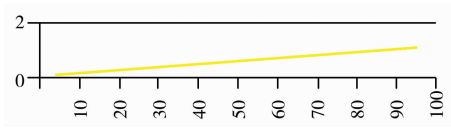


图 6　变长度的局部搜索 S_L 变化图

2.7　全局搜索策略

人工蜂群算法全局搜索是通过侦查蜂实现的，在某个解迭代 $L_{\max}$ 后仍然没有更新时，通过另外一个解替代实现跳出局部最优. 基本 ABC 算法中，侦查蜂是用一个随机产生的解表示，然而，随机产生的解往往缺失了前期迭代的有价值信息，无法利用以往搜索累积的知识提高侦查蜂的性能，从而增加了计算量. 为了解决传统 ABC 算法侦查蜂性能不足的问题，本文提出了一种基于差分进化的局部最优侦查蜂(Differential Evolutionary Local Best Scout，DELBS)策略，在提高侦查蜂性能的同时，能确保跳出局部最优，从而达到全局搜索的目的.

DELBS 策略具体描述(1)受到粒子群优化(PSO)算法的启发，为种群中每一个解 i 记录其历史迭代中搜索到的局部最优解 LB_i；(2)找到当前种群中最好和最差的局部最优解，分别记为 LB_i 和 LB_j；(3)当前种群中如果有某个解 u 迭代 $L_{\max}$ 无更新，则侦查蜂产生的公式

$$\eta_u^k = \text{Rand}() \otimes LB_i^k + (1 - \text{Rand}()) \otimes LB_j^k, 1 \leqslant k \leqslant K, \tag{19}$$

$$v_u^k = \text{Rand}() \otimes v_u^k + (1 - \text{Rand}()) \otimes \eta_u^k, 1 \leqslant k \leqslant K, \tag{20}$$

式中 $\otimes$ 表示交叉操作，η_u^k 表示挑选的两个局部最优解的交叉操作，$Rand()$是一个随机生成 0 或 1 的函数，即如果生成 0 则 η_u^k 解中第 k 辆车的客户点取自于 LB_j^k，否则取自于 LB_i^k. v_u^k 表示新的侦查蜂的第 k 辆车，其客户点取值分别来自于原解 v_u^k 和 η_u^k，选择的依据是 Rand()函数的结果值.

需要指出的是，上述操作产生的解有可能是不可行解，因而有些客户点可能重复，有些可能没有服务，修复策略如(1) 循环新解每辆车，删除重复的客户点；(2) 循环未安排服务的客户点集合，为每个客户点选择最佳插入位置，如果当前所有车辆不能为其提供服务，则新建一辆车为其服务.

2.8　IABC 算法框架

结合 上述编码、解码、局部搜索和全局搜索策略，本文提出了一种改进的 ABC 算法(Improved ABC，IABC)，分析可见，算法整体时间复杂度为 $O(n^2 m)$，算法框架如下.

Algorithm IABC
1　采用 2.5 节的初始化策略生成初始化种群.
2　Employed bee phase 雇佣蜂阶段.
for each solution i do //循环种群中每个解 i.
3　Generate a neighboring solution j by using the VLLS (cf. 2.6)for solution i//采用 2.6 解的局部搜索策略，为解 i 生成一个邻域解.
4　　if j is better than i then
5　　Replace the current solution with j //更新解 i.
6　　Update the best solution found so far with j　//更新全局最好解.
7　　Update the local best for i　//更新 i 的局部最优解.
8　　else
9　　Update the iteration time without improvement for i　//更新 i 的无更新次数.
10　　end.
11　end.
12　Onlooker bee phase 跟随蜂阶段.
for each solution i do //循环种群中每个解 i.
13　　Randomly select another solution k in the population //随机选择另一个解 k.
14　　Select the better solution between i and k as the current solution i //选择较优解.
15　　Perform steps 3-11 as listed in Employed bee phase. //执行步骤 3-11.
16　end.

Scout bee phase 侦查蜂阶段.
for each solution i do //循环种群中每个解 i.
if iteration time without improvement for i is larger than L_{max} then
    Perform global search given in 2.7 //执行 2.7 节的全局搜索.
end

3 实验分析

3.1 实验算例

经典 SOLOMN 算例,每个客户点的需求是一个值,即没有区分货物的种类、包装、容器等特性. 装配式建筑配送过程中,不同类型的预制件需要叠放在不同的容器中,为了更好地考虑实际约束,本文在 SOLOMN 经典算例的基础之上,改进了算例中客户点需求量,增加了货物种类的不同需求量. 譬如,原来的算例客户点 1 的需求量为 20,只表示出客户点需要 20 个物品,改进的算例中增加了不同物品种类,如客户点 1 的需求量为(20,60),表示该客户点需要 20 个物品 1 和 60 个物品 2. 算例的改进导致在解码过程中,需要考虑车辆当前负载是否能同时满足该客户点的所有物品需求,因而问题变得更加贴近生产实际.

扩展的 VRPTW 算例,包括 18 个算例,每个算例中包含 100 个客户点,客户点的布局分为仍然分为三大类,即聚合程度较高的 C(Clustering)系列算例、聚合程度分散的 R(Random)系列算例和聚合程度处于中间状态的 RC 系列. 客户点的需求量采用两大类物品,两类物品的需求比在 1∶1 到 1∶5 之间随机生成.

3.2 实验结果分析

本实验的参数具体包括(1) 实验终止条件,迭代 100 次;(2) 解最大迭代无更新次数 $L_{max}=10$;(3) 种群大小 100. 为了验证本文所提 ABC 算法的有效性,本文选取 SOLOMN 的 PFIH 方法,基本遗传算法 GA 作为对比算法,求解了扩展的 18 个 VRPTW 算例.

表 1 给出了算法针对 18 个 VRPTW 算例的实验对比,表中第一列给出了算例名称,第二列给出了每个算法所有对比算法获得的最好值,接下来四列展示出四种对比算法获得的每个算例的最好目标值,最后四列给出了相对于最好值,每个算法得到的均方差值,计算公式

$$dev = (f_c - f_b)/f_b \times 100\ \%. \tag{21}$$

由表可见,本文提出的 IABC 算法在求解扩展的 18 个 VRPTW 算例中(1) 获得了其中 16 个最优值,明显优于其他对比算法;(2) 通过均方差分析可见,IABC 算法除了算例 Case5、Case12 两个算例之外,其他均获得了最小值;(3) 最后一行给出的平均性能可见,IABC 算法获得了 1022.09 的平均目标值,明显优于 GA 算法获得的 1125.24,均方差的平均值可见,IABC 获得了 1.78,明显小于 GA 算法. 综上可见,所提出的算法相比于其他经典算法,具备明显的优越性.

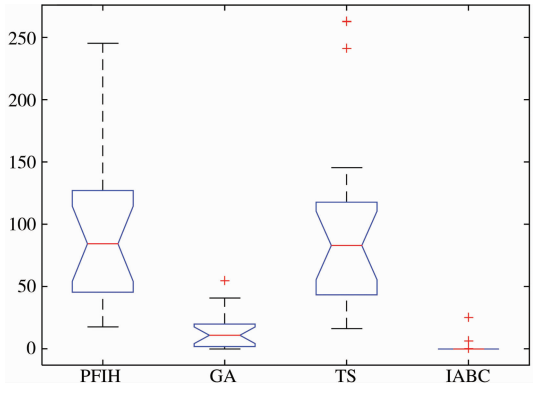


图 7 ANOVA 方差分析图

为进一步对比算法在统计意义上的优越性,本文选取三种对比算法,做了(Analysis of Variance,ANOVA)方差分析,ANOVA 用于两个及两个以上样本均数差别的显著性检验. 图 7 表明本文提出的 IABC 算法相对于其他三种对比算法表现了统计意义上的优越性.

图 8 给出了 Case1 算例的客户点服务时间 Gantt 图,图中"V_1"表示第一辆车,与其对应的每个矩形框表示一个客户点,矩形框中的编号表示客户点编号,譬如,第一辆车服务的客户点有{20,24,25,27,29,30,28,26,23,22,21},共计 11 个客户点. 每个客户点下方的两个数字表示该客户点开始服务和结束服务的时刻,例如,47 号客户点的开始服务和结束服务的时间点分别是 1125 和 1217. 经与算例数据对比可见,每个客户点的服务开始时间均在其指定的服务时间窗之内,该调度方案是可行的、有效的. 图 9 给出了算法求解 Case1 算法的收敛曲线图,由图可见,算法具备良好的收敛性能.

表 1　实验结果对比

算例	最好值	比较算法				均方差(dev)			
		PFIH	GA	TS	IABC	PFIH	GA	TS	IABC
Case1	1141.79	1653.30	1284.94	1553.92	1141.79	44.80	12.54	36.10	0.00
Case2	796.40	1642.48	1035.62	1661.88	796.40	106.24	30.04	108.67	0.00
Case3	1442.52	2831.44	1471.03	2868.27	1442.52	96.28	1.98	98.84	0.00
Case4	990.95	1857.49	1085.09	1841.05	990.95	87.45	9.50	85.79	0.00
Case5	1366.98	1723.61	1366.98	1761.56	1371.22	26.09	0.00	28.87	0.31
Case6	968.54	1489.17	1125.81	1449.04	968.54	53.75	16.24	49.61	0.00
Case7	1266.82	1844.13	1362.95	1802.43	1266.82	45.57	7.59	42.28	0.00
Case8	1146.08	1649.75	1233.32	1644.12	1146.08	43.95	7.61	43.46	0.00
Case9	975.86	1763.33	1102.84	1759.72	975.86	80.70	13.01	80.33	0.00
Case10	616.86	1118.82	740.26	1086.01	616.86	81.37	20.01	76.06	0.00
Case11	1036.14	1509.89	1120.55	1643.77	1036.14	45.72	8.15	58.64	0.00
Case12	820.19	2784.91	820.19	2797.40	1027.14	239.54	0.00	241.07	25.23
Case13	696.71	1582.37	828.46	1475.18	696.71	127.12	18.91	111.74	0.00
Case14	914.48	3157.12	1415.05	3315.38	914.48	245.24	54.74	262.54	0.00
Case15	866.83	1910.00	881.32	1887.80	866.83	120.34	1.67	117.78	0.00
Case16	632.32	1624.34	890.67	1551.97	632.32	156.89	40.86	145.44	0.00
Case17	501.38	1717.52	604.61	1819.16	501.38	242.56	20.59	262.83	0.00
Case18	1884.69	2218.69	1884.69	2192.99	2005.60	17.72	0.00	16.36	6.42
平均性能	1003.64	1893.24	1125.24	1895.09	1022.09	103.41	14.63	103.69	1.78

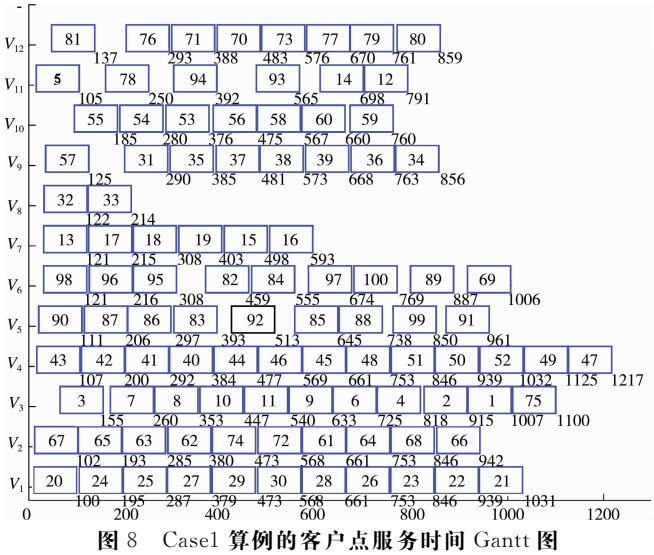


图 8　Case1 算例的客户点服务时间 Gantt 图

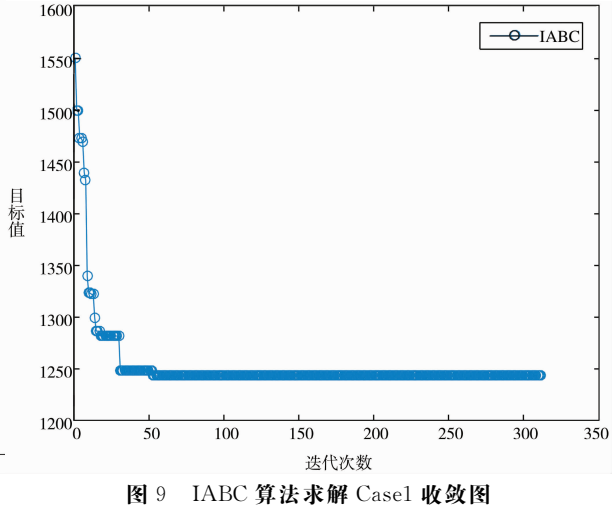


图 9　IABC 算法求解 Case1 收敛图

4　结论

本文以装配式建筑配送为研究背景，建立基于 VRPTW 的扩展模型，并采用智能优化算法进行求解. 在算法设计中，充分考虑了问题特性和目标特点，设计了雇佣蜂、跟随蜂和侦查蜂三种策略，很好的平衡了局部搜索和全局搜索. 进一步的工作主要集中于(1) 不断完善所提出的优化算法，进一步提高算法性能；(2) 应用所提出的算法求解多目标 VRPTW 问题.

参　考　文　献

[1] Dantzig G B, Ramser J H. The truck dispatching problem[J]. Management Science, 1959, 6(1): 80-91.

[2] Podgorelec V. A survey of genetic algorithms for solving multi depot vehicle routing problem[J]. Applied Soft Computing Journal, 2015, 27(C): 519-532.

[3] Neves-Moreira F, da Silva D P, Guimarães L, et al. The time window assignment vehicle routing problem with product dependent deliveries [J]. Transportation Research Part E: Logistics and Transportation Review, 2018, 116: 163-183.

[4] BRANDÃO, José. Iterated local search algorithm with ejection chains for the open vehicle routing problem with time windows[J]. Computers & Industrial Engineering, 2018, 120: 146-159.

[5] Niu Y,Yang Z,Chen P,et al. Optimizing the green open vehicle routing problem with time windows by minimizing comprehensive routing cost[J]. Journal of Cleaner Production,2018,171:962-971.
[6] Koch H,Bortfeldt A,Wäscher G. A hybrid algorithm for the vehicle routing problem with backhauls,time windows and three-dimensional loading constraints[J]. OR Spectrum,2018,18(1):1-47.
[7] Reil S,Bortfeldt A,Mönch L. Heuristics for vehicle routing problems with backhauls,time windows,and 3D loading constraints[J]. European Journal of Operational Research,2018,266(3),877-894.
[8] Liu R,Tao Y,Xie X. An adaptive large neighborhood search heuristic for the vehicle routing problem with time windows and synchronized visits[J]. Computers & Operations Research,2018,118(25):30-41.
[9] Hojabri H,Gendreau M,Potvin J Y,et al. Large neighborhood search with constraint programming for a vehicle routing problem with synchronization constraints[J]. Computers & Operations Research,2018,92:87-97.
[10] Miranda D M,Branke J, Conceição S V. Algorithms for the multi-objective vehicle routing problem with hard time windows and stochastic travel time and service time[J]. Applied Soft Computing,2018,70:66-79.
[11] Paolucci M,Anghinolfi D,Tonelli F. Field services design and management of natural gas distribution networks:a class of vehicle routing problem with time windows approach[J]. International Journal of Production Research,2018,56(3):1154-1170.
[12] Shi Y,Boudouh T,Grunder O. A hybrid genetic algorithm for a home health care routing problem with time window and fuzzy demand[J]. Expert Systems with Applications,2017,72:160-176.
[13] 刘宏令,刘俊娥. 京津冀装配式建筑预制构件配送中心选址研究[J]. 工程经济,2018,28(8):58-62.
[14] 李萍萍. 装配式 PC 构件配送成本优化研究[D]. 西安:西安建筑科技大学,2016.
[15] 彭星. 建筑施工企业部品部件物流配送中心动态选址研究[D]. 重庆:重庆大学,2017.
[16] Karaboga D,Basturk B. A powerful and efficient algorithm for numerical function optimization:artificial bee colony (ABC) algorithm[J]. Journal of Global Optimization,2007,39(3):459-471.
[17] 李俊青,王永,桑红燕,等. 人工蜂群优化及其在资源管理中的应用[J]. 聊城大学学报(自然科学版),2017,30(3):86-90.
[18] 李俊青. 求解分布式装配式建筑逆向物流问题的离散人工蜂群算法[J]. 聊城大学学报(自然科学版),2018,31(2):102-110.
[19] Li J,Duan P,Sang H,et al. An efficient optimization algorithm for resource-constrained steelmaking scheduling problems[J]. IEEE Access,2018,125(3):35-39.
[20] Li J,Sang H,Han Y,et al. Efficient multi-objective optimization algorithm for hybrid flow shop scheduling problems with setup energy consumptions[J]. Journal of Cleaner Production,2018,18:584-598.
[21] Li J,Pan Q,Duan P,et al. Solving multi-area environmental/economic dispatch by Pareto-based chemical-reaction optimization algorithm [J]. IEEE/CAA Journal of Automatica Sinica,2017,178(4):125-130.
[22] Li J Q,Wang J D,Pan Q K,et al. A hybrid artificial bee colony for optimizing a reverse logistics network system[J]. Soft Computing,2017,21(20):6001-6018.

Research on VRP with Time Window for Delivery Problem in Prefabricated Building

LI Jun-qing[1] SONG Mei-xian[1] DENG Jia-wen[1] HAN Yun-qi[2]

(1. School of Computer Sciences,Liaocheng University,Liaocheng 252059,China;2. School of Information Science and Engineering,Shandong Normal University,Jinan 250014,China)

Abstract During recent years,the vehicle routing problem with time window (VRPTW) has gained more and more researches and focuses. Additionally,the prefabricated building has become a typical building type,where the delivery process of the prefabricated precast units can be considered as a complex engineering optimization problem. Considering the background of the prefabricated building,in this study,a literature review of the recent development of VRPTW is firstly presented. Then,an extended VRPTW model is formulated. Next,based on the intelligent optimization algorithms,a novel heuristics is designed to solve the considered problem. Based on the classical SOLOMN instances,eighteen instances with different features are randomly generated to test the proposed algorithm. Then,the efficiency and effectiveness of the proposed algorithm is verified compared with other efficient algorithms.

Key words vehicle routing problem; prefabricated building delivery; intelligent optimization algorithm; prefabricated units; time window