

基于混合蚁群算法的动态设施规划系统研究

李 辉¹, 黄国文¹, 齐二石¹

(1. 天津大学管理与经济学部, 天津 300072; 2. 天津大学管理与经济学部, 天津 300072;
3. 天津大学管理与经济学部, 天津 300072)

摘要:设施规划问题主要研究生产设备的布局规划,从而减小厂区内的物料搬运成本。一个有效的设施规划有利于生产过程中整体运作效率的提高。随着市场竞争的日趋激烈,市场环境处于不断的变化之中。制造企业需不断对设施布局进行重新规划来适应不断变化的市场环境对产品需求量的影响,并达到降低成本的目的。这一问题需要用动态设施规划(DFLP)的方法来解决。本文提出了一种改进的混和蚁群算法(HACO)来解决带有财务预算约束的动态设施规划问题,并将此方法与其他一些典型的启发式算法进行了对比分析。结果表明,本文提出的HACO算法是求解带有财务预算约束的DFLP问题的一种有效的方法。

关键词:动态设施规划;混合蚁群算法;财务预算约束

中图分类号:F273 **文献标识码:**A **文章编号:**1004-6062(2014)01-0110-09

0 引言

当今世界,全球的竞争越来越体现在经济和科技实力的竞争,而技术创新则日益成为促进经济增长和提高科技竞争力的关键。技术获取模式作为企业技术战略的重要一环,对企业长远发展意义重大。

设施规划问题主要研究生产设备的布局规划,从而减小厂区内的物料搬运成本。据统计资料表明,一个生产车间的物料搬运成本占整个生产运作成本的20%~50%,占产品生产总成本的15%~70%^[1]。在一个生产车间中,一个有效的设施规划能够调整设备间的物流量,从而使每个设备在正确的时间得到正确数量的物料,这样既可以减少在制品的库存量,又能够防止厂房中机器设备的过度使用,减少物流成本。一个有效的设施规划有利于生产过程中整体运作效率的提高^[2]。

物流成本主要由物料在各设备间的流动量以及各设备间的相对距离来决定。如果各设施间的物流量自始至终都是固定不变的,这样的设施规划问题称为静态设施规划问题(Static Facility Layout Problem, SFLP)^[3]。相反,如果各设备间的物流量在不同的阶段内均有所变化,静态设施规划的方法就显得无能为力了,为了维持、甚至提高制造系统的效率,就需要一种新的规划方式,即根据不同时期生产系统中物流量的变化重新规划生产系统或调整生产系统结构,这就是动态设施规划问题(Dynamic Facility Layout Problem, DFLP)。近几年,有不少学者对动态设施规划进行了研究。动态设施规划与静态设施规划不同的是,它考虑的是一个多阶段的设施规划问题。不同企业的生产制造系统划分阶段有不同的

标准,可以以星期、月或年为单位划分。在一个阶段内,各设备间的物流量假定是恒定不变的。但是在不同的阶段,由于生产系统发生了变化,各设备间的物流量会有所改变,这就需要在每个阶段初期对设施布局进行重新设计^[1]。导致设施间的物流量发生变化的因素主要有^[4]:

- ①改变现有产品的设计;
- ②产品种类的增加或减少;
- ③现有生产设备的更换;
- ④产品生命周期的变化;
- ⑤产品生产计划和产量的改变。

因此,有必要设计一个柔性生产系统来应对以上因素对设备间物流量变化的影响。据统计,有1/3的美国企业平均每两年都会对生产系统做一次大的调整^[5]。传统上,设施规划的有效性与各设备间的物流量有关。物料搬运成本的最小化通常被当作评价设施规划有效性的标准。然而对生产系统进行调整或重新规划也需要一定的费用。因此,就需要在物料搬运成本和生产系统的重新规划成本之间取得一个权衡。这就是动态设施规划需解决的主要问题。

许多学者都对设施规划进行过研究^[6]。但是大多数文献都是针对静态设施规划问题进行的研究。人们开始关注动态设施规划问题是近些年的事。Rosenbaltt第一个对动态设施规划问题进行了建模和求解^[7]。他是用动态规划的模型对DFLP问题进行了最优化求解。在求解过程中,动态规划模型中的每一个阶段与设施规划中的不同阶段相对应。相比于SFLP问题,DFLP问题的求解会更加复杂。对于一个有N个设备T个阶段的生产系统来说,将会有 $(N!)^T$ 种布局

收稿日期:2011-10-24 修回日期:2012-05-22

基金项目:国家自然科学基金面上资助项目(70671072)

作者简介:李辉(1983—),男,天津人,天津大学管理学院博士研究生(在读),研究方向:工业工程、精益设计。

方案。因此,只有在问题规模不很大的时候才有可能在合理的运算时间内求出最优解。对于规模较大的 DFLP 问题,常用启发式算法进行求解。Lacksonen 和 Ensore 将一系列的静态设施规划算法进行了修正,建立了一个改进的二次规划模型^[8]。Urban 提出了一种基于定量布置技术(Computerized Relationship Facilities Technique, CRAFT)的启发式算法^[9]。Balakrishnan 提出了两种启发式算法改进了 Urban 的最速下降成对交换法^[10]。Conway 和 Venkataramanan 应用改进的遗传算法解决 DFLP 问题^[11]。Balakrishnan 提出了一种混合遗传算法^[12]。Kaku 和 Mazzola 提出了一种基于禁忌搜索法的启发式算法解决 DFLP 问题^[13]。模拟退火法也同样被用于解决 DFLP 问题^[14]。Erel 等人使用动态规划和模拟退火法相结合的方法对 DFLP 问题提出了一系列启发式算法^[15]。Kochhar 和 Heragu 研究了多层动态设施规划问题^[16]。Balakrishnan 和 Cheng 对现有的 DFLP 问题的各种算法进行了比较,并详细分析了各种算法的优劣性^[17]。

之前的文献很少考虑财务预算方面的约束。然而,我们应注意到,如果考虑到了财务预算方面的限制,某些阶段中的重新规划方案就变得不可行了。因此,财务预算的约束对动态设施规划方案的选择会产生显著的影响^[18]。

1 DFLP 问题描述和数学建模

SFLP 问题假设各设备间的物流量是恒定不变的,因此它的规划方案是要使各设备间总的物料搬运费用最小化。而 DFLP 问题不仅要在每一个阶段确定一个静态布局方案,而且还要决定在下一个阶段是否进行重新规划来改变生产系统的现有布局以提高生产系统的运作效率。如果对生产系统进行重新布局规划,就会产生重新布局规划的成本。这部分成本是由于设施的移动、产量的变化或对相关人员或器材的特殊需求等原因导致的^[19]。如果重新规划的成本比较低,我们就倾向于在每个阶段重新规划生产系统以维持它的高效率。相反的,如果重新规划的费用过高,我们就有可能维持现有规划方案不变。本文考虑了财务预算对 DFLP 的影响,在某些阶段,由于财务预算的限制,即使重新规划生产系统是有利的,也无法真正实施。

本文对 DFLP 问题的假设如下:

- ①各设备间的物流量随每个阶段的变化而变化,而在每个阶段内物流量恒定不变;
- ②每个设备具有相同的形状和面积;
- ③生产系统中设施能摆放的位置是确定的,并且和设施的数量相同;
- ④各位置间的相对距离是预先确定的。

DFLP 问题的总成本是由各设备间的总物料搬运成本与各时期生产系统重新规划成本组成。因此,DFLP 的目标函数通常被定义成各设备间的总物料搬运成本与各时期生产系统重新规划成本在各个阶段总和的最小化^[7,8,20]。本文的 DFLP 数学模型如下所示:

$$\text{Min } Z = \sum_{t=1}^T \sum_{i=1}^N \sum_{j=1}^N \sum_{k=1}^N \sum_{l=1}^N c_{ijk} f_{ijk} d_{ijl} X_{ijl} X_{klt} + \sum_{t=2}^T \sum_{i=1}^N \sum_{j=1}^N \sum_{l=1}^N A_{ijl} Y_{ijl} \quad (1)$$

$$\text{s. t. } \sum_{i=1}^N X_{ijl} = 1, j = 1, \dots, N, T = 1, \dots, T \quad (2)$$

$$\sum_{j=1}^N X_{ijl} = 1, i = 1, \dots, N, T = 1, \dots, T \quad (3)$$

$$Y_{ijl} = X_{(i-1)ij} X_{ijl}, i, j, l = 1, \dots, N, t = 2, \dots, T \quad (4)$$

$$X_{ijl} \in \{0, 1\}, i, j = 1, \dots, N, t = 1, \dots, T \quad (5)$$

$$Y_{ijl} \in \{0, 1\}, i, j, l = 1, \dots, N, t = 2, \dots, T \quad (6)$$

$$\sum_{i=1}^N \sum_{j=1}^N \sum_{l=1}^N A_{ijl} Y_{ijl} \leq LB_{t-1} + B_t, t = 2, \dots, T \quad (7)$$

其中, i, k 表示生产系统中的设备, j, l 表示生产系统中设备能够摆放的位置, c_{ijk} 表示在阶段 t 中 i, k 两设备间的单位物料搬运成本(两设备间一单位物料搬运一单位距离所花费的成本), f_{ijk} 在阶段 t 中 i, k 两设备间的物流量, d_{ijl} 表示在阶段 t 中 j, l 两位置的距离, A_{ijl} 表示在阶段 t 中将设备 i 从位置 j 移动到位置 l 所需要的固定成本, LB 表示上一阶段剩余财务预算, B 表示当前阶段财务预算。

在以上模型中,公式(1)即目标函数,是要求取各阶段的物料搬运成本以及生产系统重新规划成本之和的最小值;公式(2)说明在各个阶段,每一个设备只能放在一个位置;公式(3)说明在各个阶段,每一个位置只能容纳一个设备;公式(4)说明只有当设备改变位置时,才会产生重新规划成本;公式(5)说明 X_{ijl} 属于 0,1 变量,当在阶段 t 中设备 i 处于位置 j 上时, $X_{ijl} = 1$, 否则 $X_{ijl} = 0$;公式(6)说明 Y_{ijl} 也属于 0,1 变量,当在阶段 t 的开始设备 i 从位置 j 移动到了位置 l 时, $Y_{ijl} = 1$, 否则 $Y_{ijl} = 0$ 。公式(7)说明生产系统重新规划成本要满足预算要求。

对于 DFLP 问题,我们假设在每个阶段内各设备的物流量是恒定不变的。因此,在每个阶段内的设施规划问题均可看作是一个 SFLP 问题。我们用 π_t 表示在阶段 t 中生产系统中 N 个设备、 N 个位置的布局方案,即 $\pi_t = (\pi_{t1}, \pi_{t2}, \dots, \pi_{tN})$ 。其中 π_{ti} 表示在阶段 t 中位置 i 上的设备 ($i = 1, 2, \dots, N$)。因此,DFLP 问题的一个求解方案便可表示成

$$\pi = \{\pi_1, \pi_2, \dots, \pi_T\} = \{(\pi_{11}, \pi_{12}, \dots, \pi_{1N}), (\pi_{21}, \pi_{22}, \dots, \pi_{2N}), \dots, (\pi_{1T}, \pi_{2T}, \dots, \pi_{NT})\}$$

以图 1 为例,图 1 描述了一个有 6 个设备 3 阶段的 DFLP 问题示例。在阶段 1,设施 6,4,2,1,5,3 被分别安置在 1,2,3,4,5,6 号位置。阶段 1 的布局可表示成 $\pi_1 = (6,4,2,1,5,3)$ 。阶段 2 和阶段 3 以此类推。因此此示例的 3 个阶段总的规划方案可表示成:

$$\pi = \{(6,4,2,1,5,3), (5,4,2,1,6,3), (5,4,3,1,6,3)\}$$

在阶段 2 中,设备 5 和设备 6 调换了位置,因此阶段 2 的重新规划成本就是将设备 6 从位置 1 移到位置 5 的费用加上将设备 5 从位置 5 移到位置 1 的费用。由于 2,3 两阶段的布局没有改变,所以在阶段 3 并不产生重新规划成本。而且

每个阶段的物料搬运成本也没有变化。这样便可计算出 DFLP 整体规划的总成本。

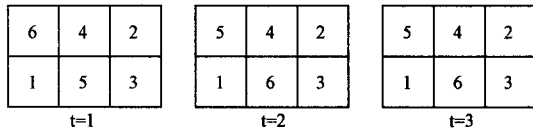


图 1 一个 6 设施 3 阶段的 DFLP 问题示例

2 混和蚁群算法对 DFLP 问题的求解

2.1 蚁群算法描述

蚁群算法 (Ant Colony Optimization Algorithm, ACO) 是近年来新发展起来的一种启发式算法,由 Dorigo 等学者在 20 世纪 90 年代首先提出^[21],而后又进行了一系列改进。由于蚁群算法采用分布式并行计算机制,具有较强的鲁棒性,容易与其他算法结合等优点,一经提出,立即受到各领域学者的重视。近几年,蚁群算法被应用于加工车间调度问题、图形着色问题、二次分配问题、车辆路径问题等一系列 NP 难问题^[22]。

蚁群算法模仿了自然界中蚂蚁觅食的行为,主要通过蚂蚁群体之间的信息传递而达到寻优的目的。蚂蚁在移动过程中会释放出一种称作“信息素”的化学物质。信息素浓度越高,对蚂蚁的吸引力就越强,因此蚂蚁更倾向于选择信息素浓度高的路径。一开始每个蚂蚁并不知道食物在何处,只是在本身所能见到的局部范围内搜索,并以一定的概率向其他蚂蚁留下的信息素浓度高的方向移动,同时自己也释放信息素。如果许多蚂蚁都经过了同一条路径,则此路径上的信息素浓度便会升高,后来的蚂蚁就更倾向于选择这条路径。同时所有蚂蚁释放的信息素将以一定速率挥发,因此经过一段时间后,最短路径上的信息素浓度会越来越,从而形成一种正反馈,因此到最后所有的蚂蚁都会选择最短的路径去寻找食物。这一过程如图 2 所示。

为了能够有效的应用蚁群算法解决 DFLP 问题,我们可将蚁群算法解空间看作一种网状结构。图 2 描述了一个 2 阶段 3 设备的网状结构可行解示意图。在这一结构中假设有一只蚂蚁(即一个可行解)。在第一阶段,蚂蚁所经过的位置 1-2-3 中的设施为 2-1-3,。而在第二阶段,蚂蚁所经过的位置 1-2-3 中的设施为 3-2-1。因此,这两个阶段的可行解便可表示成一个字符串的形式:2-1-3-3-2-1。这一可行解也可表示成结构书的形式如图 3 所示。

2.2 蚁群算法求解 DFLP 问题的步骤

蚁群算法的核心内容包括以下三部分:(1) 选择机制:某条路径上留下的信息素越多,此路径被选择的概率越大;(2) 更新机制:某条路径上的信息素浓度会随着蚂蚁经过数量的增加而变大,同时信息素也会逐渐的挥发;(3) 协调机制:蚂蚁之间通过信息素交换信息并相互影响。

求解过程中所用字母的含义如下:

M : 蚁群规模(蚂蚁的数量)

m : 蚂蚁序号 ($m = 1, \dots, M$)

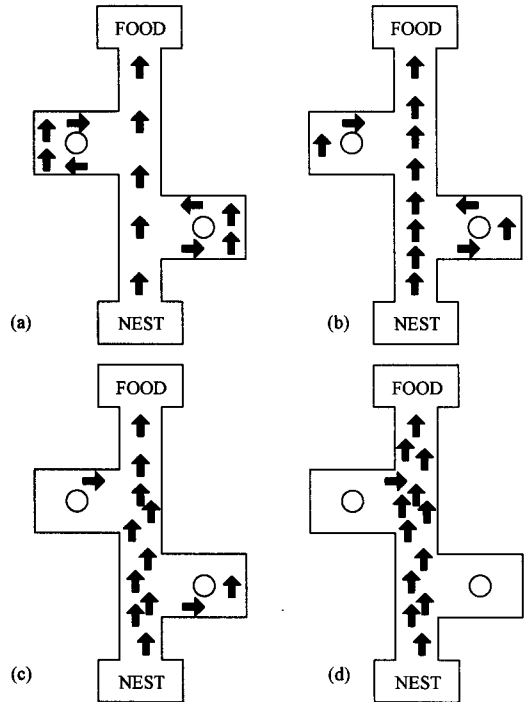


图 2 蚂蚁寻找最短路径过程示意图:(a) 初态;(b) 阶段一;(c) 阶段二;(d) 终态:蚂蚁集中于最短路径

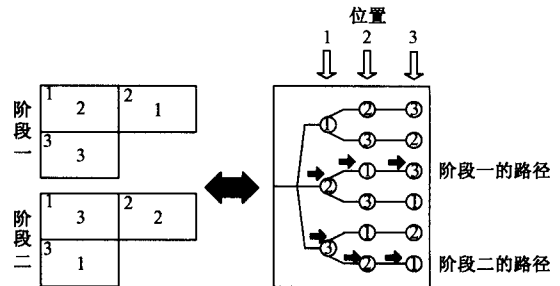


图 3 DFLP 问题蚁群算法解空间的网络表示

N : 设施(位置)的数量

i, z : 生产系统中的设备 ($i, z = 1, \dots, N$)

j : 生产系统中的位置 ($j = 1, \dots, N$)

S : 阶段总数

t : 阶段

F : 目标函数

BL_t : 阶段 t 的最优布局方案

r : 失败迭代计数器,记录连续迭代连续失败的次数

R : 失败迭代的上限

P_{ij}^t : 信息素浓度(在阶段 t 中将设施 i 放置在位置 j 中的倾向性)

Q : 信息素更新参数

W : 交换次数。

α : 信息素挥发率

u : ACO 流程迭代次数

U : ACO 流程迭代上限

用蚁群算法求解 DFLP 问题的步骤如下:

I. 确定初始解

令 F_{best} = 一个很大的数。从第一只蚂蚁开始 ($m = 1$) 对每一只蚂蚁重复进行如下操作(重复 M 次):

步骤 1: 随机确定第一阶段的设备布局方式, 即将所有的设备随机排列。令之后的每一阶段的设备布局方式不变, 因此其重新规划成本为 0 ($\pi_1 = \pi_2 = \dots = \pi_s$)。

步骤 2: 改进过程。令 $r = 0$ 。计算初始布局方案的目标函数 (F_m)。随机选择一个阶段, 在此阶段中随机选择 2 个设备进行交换。检验是否满足预算约束条件。若满足条件, 计算交换设备后的目标函数 (F_m')。若 $F_m > F_m'$, 则接受改变后的布局方式, 并令 $r = 0$ 。如果 $F_m < F_m'$ 或不满足预算约束条件, 就要重新选择两个设备进行交换, 并令 $r = r + 1$ 。

步骤 3: 更新最优解。如果第 m 只蚂蚁的目标函数 (F_m) 优于现阶段最优解, 即 $F_m < F_{best}$, 则要更新最优解 ($F_{best} = F_m$, $BL_t = \pi_t, t = 1, \dots, S$), 否则, 最优解不更新。

步骤 4: 更新蚂蚁数量 ($m = m + 1$), 返回步骤 1。

根据以上步骤, 可确定问题的一个初始解。此时各路径上的信息素浓度为 $P_{ij} = P_{ij} + Q/F_{best}$ ($i = 1, \dots, N, t = 1, \dots, S, Q$ 表示所规定的目标函数的上限)。

II. ACO 流程

设置参数 $m = 1, r = 0$ 。重复操作以下步骤, 直到迭代次数达到迭代上限为止:

步骤 1: 更新初始解(令 $\pi_t = BL_t, t = 1, \dots, S$)。

步骤 2: 对每一只蚂蚁重复以下操作:

步骤 2.1: 交换过程。重复以下交换过程 W 次:

首先, 随机选择一个阶段, 并在所选阶段中随机选择一个设备 u 。然后, 再选择一个设备 v 。选择设备 v 的概率与设备 u 和设备 v 的相关信息素浓度 ($P_{u,lc(v),t} + P_{v,lc(u),t}$) 成正比 ($lc(u)$ 表示设备 u 的位置)。最后, 交换所选两设备的位置。需要强调一点: 设备 v 与设施 u 可以相同。当所选两设备相同时, 说明设备的位置没有改变。

根据信息素的浓度选择设备 v 的具体操作如下: 令

$$tf = \sum_{i=1}^N P_{u,lc(i),t} + P_{v,lc(i),t} \quad (8)$$

$$u = 1 / (tf + N\beta) \quad (9)$$

$$cp(i) = [(P_{u,lc(i),t} + P_{v,lc(i),t}) + \beta]u, i = 1, \dots, N \quad (10)$$

$$\sum_{i=1}^N cp(i) = 1 \quad (11)$$

$$kp(i) = kp(i-1) + cp(i), i = 1, \dots, N (kp(0) = 0, kp(N) = 1) \quad (12)$$

任意取 $q \in (0, 1)$, 选择满足以下不等式的第 i 个设备作为设施 v :

$$kp(i-1) \leq q < kp(i) \quad (13)$$

步骤 2.2: 改进过程。与“I. 确定初始解”中的“改进过程”步骤完全相同。

步骤 2.3: 更新信息素。首先将信息素进行挥发 ($P_{ij} = P_{ij} * \alpha, i = 1, \dots, N, j = 1, \dots, N, t = 1, \dots, S$), 再聚集信息素 ($P_{ij} = P_{ij} + Q/F_{best} * \alpha, i = 1, \dots, N, j = 1, \dots, N, t = 1, \dots, S$)。

步骤 2.4: 更新最优解。如果第 m 只蚂蚁的目标函数优于当前最优解 ($F_m < F_{best}$), 则要将最优解进行更新 ($F_{best} = F_m, BL_t = \pi_t, t = 1, \dots, S$)。这样, 下一只蚂蚁就可以使用改进后的最优解作为其初始解了。再次挥发信息素 ($P_{ij} = P_{ij} * \alpha, i = 1, \dots, N, j = 1, \dots, N, t = 1, \dots, S$) 和聚集信息素 ($P_{ij} = P_{ij} + Q/F_{best}, i = 1, \dots, N, j = 1, \dots, N, t = 1, \dots, S$)。由于消除了参数 α 对信息素增加的影响, 此时信息素浓度增加的更多。这样一来最优解被选中的概率会增加。最后令 $r = 0$ 。

步骤 2.5: 若蚂蚁数量 $m = M$ 则继续执行以下步骤, 否则令 $m = m + 1$, 返回步骤 2.1。

步骤 3: 如果最优解没有更新, 则令 $r = r + 1$ 。如果 $r = R$, 则清空所有信息素 ($P_{ij} = 0, i = 1, \dots, N, j = 1, \dots, N, t = 1, \dots, T$) 并重新聚集信息素 ($P_{ij} = P_{ij} + Q/F_{best}, i = 1, \dots, N, j = 1, \dots, N, t = 1, \dots, S$)。

步骤 4: 返回步骤 1, 进行下一轮迭代。

以上 ACO 算法的流程图如图 4 所示。

根据以往实验结论, 以上步骤所需参数的取值如下: $R = (N * S) / 2, W = (N * S) / 2, \alpha = 0.5, \beta = 1$ 。

2.3 改进的混和蚁群算法 (HACO) 求解 DFLP 问题的步骤

本文在简单蚁群算法的基础上加入了模拟退火法 (SA), 形成了混和蚁群算法 (Hybrid Ant Colony Optimization Algorithm, HACO), 从而改进了原始蚁群算法的性能。

模拟退火法是一种解决混合优化问题的随机方法, 它模拟了金属煅烧退火的过程。在这一过程中, 一个金属固体被加热到其融化, 而后它的温度不断降低直到达到一种最低能量的状态 (或称基态)。如果初始温度不够高或温度下降的过快, 固体达到基态时就会产生多种缺陷。Kirkpatrick 等学者第一次运用 SA 解决混合优化问题^[23]。Wilhelm 和 Ward 运用 SA 解决 SFLP 问题^[24]。Baykasoglu 和 Gindy 第一次运用 SA 解决 DFLP 问题^[25]。

本文的混合蚁群算法就是用模拟退火法来替换上述蚁群算法中的两次“改进过程”。本文中的模拟退火法的步骤如下:

步骤 1: 定义 SA 参数: 令 T_0 = 初始温度; T = 当前温度; γ = 冷却率; $AM = 2(N)(T^2)$ = 每一个温度下的最大设备交换次数; $T_{min} = 0.01$ = 最低温度。

步骤 2: 初始化温度变化计数器 $k = 1$ 。

步骤 3: 对于初始布局方案 $\pi = \{\pi_1, \pi_2, \dots, \pi_s\}$, 计算其目标函数 $f(\pi)$, 令 $BS = \pi, F_{best} = f(\pi)$ 。

步骤 4: 如果当前温度 $T \leq T_{min}$, 即停止迭代, 并返回 $BS = \pi, F_{best} = f(\pi)$ 。否则, 初始化设施交换次数计数器 $a = 0$, 并设置当前温度 $T = T_0 \gamma^{k-1}$ 。

步骤 5: 任选一阶段 t , 在阶段 t 中任意交换两个设施的位置, 得到一新的规划方案 π' 。

步骤 6: 令 $a = a + 1$, 计算目标函数的变化量 $\Delta f = f(\pi') - f(\pi)$, 如果 $\Delta f < 0$, 或 $\Delta f > 0$ 且 $x = \text{random}(0, 1) < P(\Delta f) = \exp(-\Delta f/S)$, 则接受这一新方案, 令 $\pi = \pi'$ 。若 $F_{best} > f(\pi)$, 则令 $F_{best} = f(\pi)$, 否则, 拒绝这一新方案。

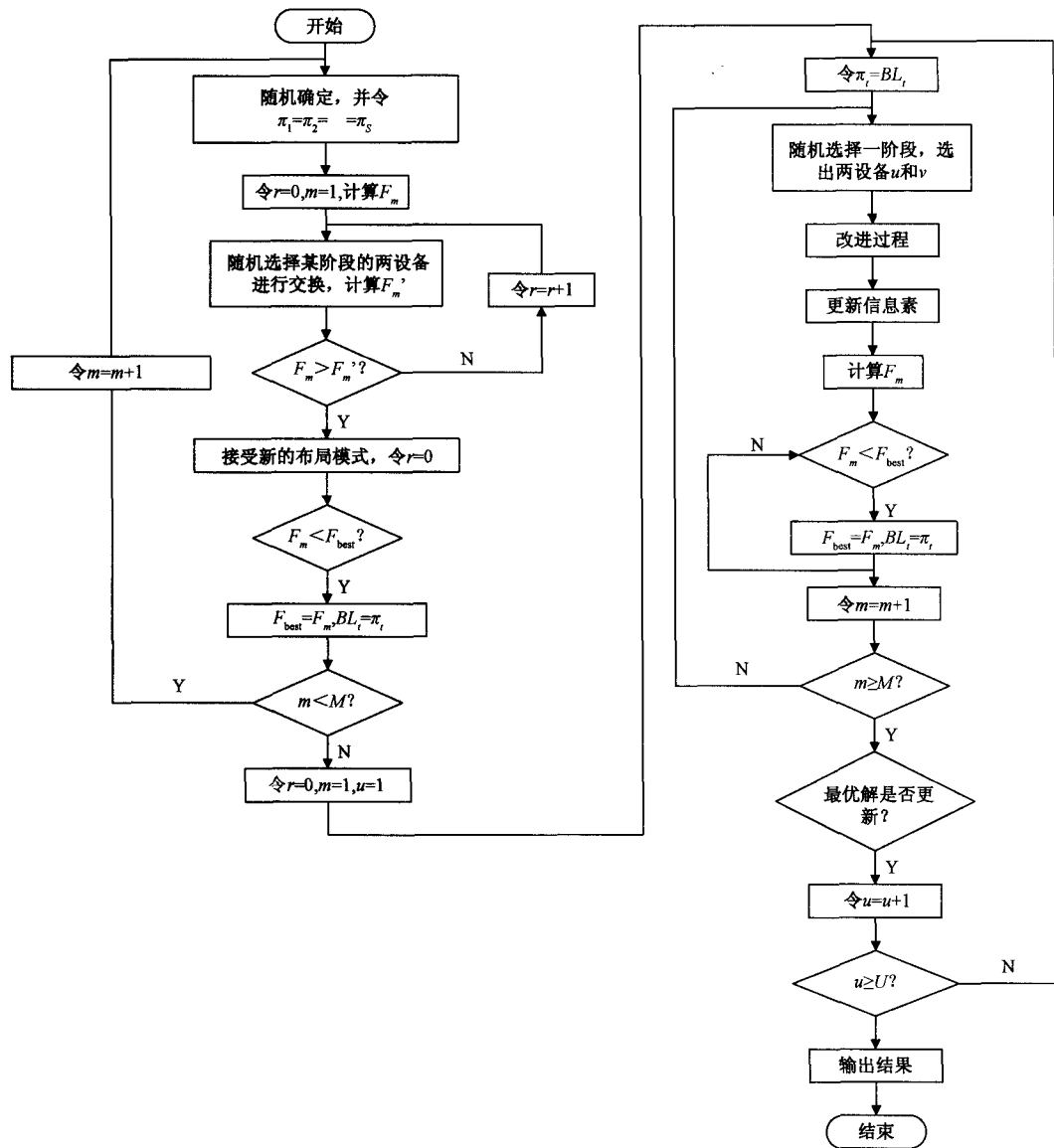


图 4 ACO 算法流程图

步骤 7: 如果 $a \geq AM$, 更新 $k = k + 1$, 返回步骤 3; 否则, 返回步骤 4。

以上步骤的流程图如图 5 所示。

以上模拟退火法中的各个参数值的确定方法如下: 初始温度 T_0 由公式 $P(\Delta f) = \exp(-\Delta f/T_0)$ 来确定, 根据以往的实证研究^[3], 一般设 $\Delta f = 0.10 f(\pi)$, $P(\Delta f) = 0.25$, 因此, $T_0 = -0.10 f(\pi) / \ln(0.25)$ 。我们分别设 $\gamma = 0.99$, $AM = 2(N)(S^2)$ 。

3 数据分析

在这一节中, 我们利用 Balakrishnan 和 Cheng 的文献资料^[26]中的 48 组检测数据来对本文所提出的简单蚁群算法 (ACO) 和改进的蚁群算法 (HACO) 的运算效率进行检验。这 48 组数据分三种类型, 分别是每组 6 台设备、每组 15 台设备和每组 30 台设备, 每种类型各占 16 组。在每种类型的

数据中, 阶段数为 5 和阶段数为 10 的数据各占一半, 均为 8 组。

表 1 至表 3 描述了本文提出的简单蚁群算法和改进的混和蚁群算法与其他一些用于解决 DFLP 问题的典型的启发式算法进行了对比, 其他启发式算法包括 Baykosaglu 和 Gindy 提出的模拟退火法 (SA)^[14], Balakrishnan 提出的混合遗传算法 (GA)^[12], Erel 提出的动态规划方法 (DP)^[15]。为了提高算法的精确度, 本文将每一个参与检验的算法对每一组检验数据均计算 3 次, 并取其最优解。表 1 至表 3 显示了每一种算法对每一个数据经过 3 次计算后的最优解。简单蚁群算法和混合蚁群算法的数据分别填入“ACO”和“HACO”两列中, 而模拟退火法、混合遗传算法以及动态规划方法所得到的最优解数据分别填入“SA”, “GA”和“DP”列中。三个表格中的粗体数字均表示每一组数据达到最优解的计算结果。

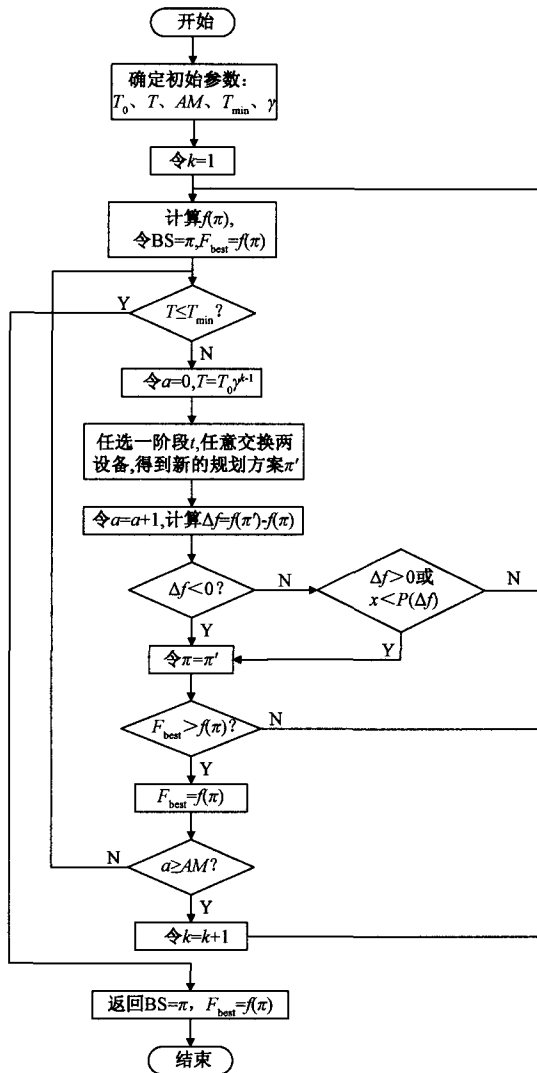


图5 SA算法流程图

由表1可知,在6台设备的数据中,对于阶段数为5的8个数据(数据1~8),本文提出的ACO和HACO两种蚁群算法均得到了最优解。其次是GA和DP算法,分别得到了7次和6次最优解,SA算法最少,仅得到2次最优解;对于阶段数为10的8数据(数据9~16),ACO和HACO两种蚁群算法全部得到最优解,其次是DP算法,得到6次最优解,而SA算法没有出现最优解。因此,对于表1中的数据,ACO和HACO是两个最优算法。

由表2可知,在15台设备的数据中,对于阶段数为5的8个数据(数据17~24),HACO算法得到了4次最优解,而ACO算法仅得到2次最优解,其他算法(SA、GA和DP)均没有得到最优解;对于阶段数为10的8个数据(数据25~32),HACO算法得到了5次最优解,而ACO算法仅得到1次最优解,其他三种算法没有得到最优解。由此可见,在15台设备的数据中,HACO算法明显优于其他算法。

由表3可以看出,在30台设备的数据中,对于阶段数为5的8个数据(数据33~40),HACO算法得到了6次最优解,其次是ACO和GA两种算法,均得到了1次最优解,而其他算法没有得到最优解;对于阶段数为10的8个数据(数据41~48),HACO算法得到了4次最优解,ACO和GA算法各得到2次最优解,而其他算法均无最优解。由此可见,在30台设备的数据中,HACO算法仍然是最优算法,其次是ACO和GA两种算法。

通过以上分析可以看出,在这48个设施规划数据中,SA、GA、DP和ACO方法分别得到了2次、13次、12次和22次最优解,而HACO方法则得到了36次最优解,其最优解的数量远远多于其他方法。从表1至表3中的数据我们可以看出,本文所提出的HACO方法无论是解决小规模设施规划问题还是大规模问题,均能表现出良好的性能。而其他方

表1 6台设备的设施规划问题的计算结果

数据	阶段数	最优解	SA	GA	DP	ACO	HACO
1	5	106,419	107,249	106,419	106,419	106,419	106,419
2		104,834	105,710	104,834	104,834	104,834	104,834
3		104,320	104,800	104,320	104,320	104,320	104,320
4		106,339	106,515	106,515	106,509	106,399	106,339
5		105,628	106,282	105,628	105,628	105,628	105,628
6		103,985	103,985	103,985	103,985	103,985	103,985
7		106,439	106,447	106,439	106,447	106,439	106,439
8		103,771	103,771	103,771	103,771	103,771	103,771
9	10	214,313	215,200	214,313	214,313	214,313	214,313
10		212,134	214,713	212,134	212,134	212,134	212,134
11		207,987	208,351	207,987	207,987	207,987	207,987
12		212,530	213,331	212,741	212,741	212,530	212,530
13		210,906	213,812	210,944	211,022	210,906	210,906
14		209,932	211,213	210,000	209,932	209,932	209,932
15		214,252	215,630	215,452	214,252	214,252	214,252
16		212,588	214,513	212,588	212,588	212,588	212,588

表 2 15 台设备的设施规划问题的计算结果

数据	阶段数	最优解	SA	GA	DP	ACO	HACO
17	5	480,453	501,447	484,090	482,123	481,395	480,453
18		484,761	506,236	485,352	485,707	484,761	484,879
19		488,748	512,886	489,898	491,310	490,899	490,398
20		484,446	504,956	484,625	486,851	485,561	484,446
21		487,722	509,636	489,885	491,178	489,012	489,206
22		486,685	508,215	488,640	489,847	487,417	486,965
23		486,853	508,848	489,378	489,155	486,853	486,853
24		491,016	512,320	500,779	493,577	493,963	491,016
25	10	980,351	1,017,741	987,887	983,070	982,208	980,351
26		978,271	1,016,567	980,638	983,826	981,802	980,197
27		978,027	1,021,075	985,886	988,635	978,027	984,065
28		974,694	1,007,713	976,025	976,456	978,987	975,854
29		979,196	1,010,822	982,778	982,893	980,611	979,196
30		971,548	1,007,210	973,912	974,436	973,508	971,548
31		980,752	1,013,315	982,872	982,790	985,659	980,752
32		985,707	1,019,092	987,789	988,584	987,986	985,707

表 3 30 台设备的设施规划问题的计算结果

数据	阶段数	最优解	SA	GA	DP	ACO	HACO
33	5	576,886	604,408	578,689	579,741	578,854	580,240
34		570,349	604,370	572,232	570,906	570,349	570,349
35		576,053	603,867	578,527	577,402	578,152	578,176
36		566,777	596,901	572,057	569,596	569,694	566,777
37		558,353	591,988	559,777	561,078	560,433	558,353
38		566,792	599,862	566,792	567,154	569,725	566,792
39		567,131	600,670	567,873	568,196	570,899	567,131
40		575,280	610,474	575,720	575,273	576,980	575,280
41	10	1,166,164	1,223,124	1,169,474	1,171,178	1,166,164	1,166,164
42		1,168,878	1,231,151	1,168,878	1,169,138	1,169,407	1,168,878
43		1,166,366	1,230,520	1,166,366	1,165,525	1,166,916	1,166,720
44		1,148,202	1,200,613	1,154,192	1,152,684	1,190,140	1,148,202
45		1,128,855	1,210,892	1,133,561	1,128,136	1,128,855	1,133,664
46		1,141,344	1,221,356	1,145,000	1,143,824	1,153,982	1,149,040
47		1,140,773	1,212,273	1,145,927	1,142,494	1,152,805	1,140,773
48		1,166,157	1,231,408	1,168,657	1,167,163	1,172,787	1,166,157

法在解决大规模设施规划问题时较 HACO 方法则有较大差距。可证明本文所提出的 HACO 方法是解决带有财务预算约束的动态设施规划问题的一种有效方法。

4 结论

本文构建了带有财务预算约束的动态设施规划模型,并提出了两种蚁群算法来对此模型进行求解。先用简单蚁群算法对此模型进行求解。而后再在简单蚁群算法的基础上

加以改进,融入了模拟退火法,形成了混和蚁群算法,从而提高了蚁群算法的计算性能。利用 Balakrishnan 和 Cheng 的 48 组监测数据^[26],将本文提出的混合蚁群算法与其他一些用于解决 DFLP 问题的启发式算法进行比较,实践证明,本文所提出的混合蚁群算法能够有效的解决带有财务预算约束的 DFLP 问题。

本文认为,在本文研究的基础上,还有一些后续问题值得进一步的研究和探讨。这些问题包括:

(1) 本文在模型构建中, 仅以成本最小化作为模型目标函数。今后的研究可考虑建立并求解多目标模型;

(2) 本文仅研究了生产车间的单层动态设施规划问题, 今后可进一步研究多层生产车间的设施规划问题;

(3) 在本文提出的算法的基础上, 研究是否还有其他更加有效的算法解决此类问题。

参 考 文 献

- [1] Tompkins JA, White JA, BozerYA, Frazelle EH, Tanchoco JMA, Trevino J. Facilities planning [M]. NewYork: Wiley; 1996. p. 137 ~ 285.
- [2] Berna Haktanirlar Ulutas, A. Attila Islier. A clonal selection algorithm for dynamic facility layout problems [J]. Journal of Manufacturing Systems. 2009, 28: 123 ~ 131.
- [3] Alan R. McKendall Jr., Jin Shang. Hybrid ant systems for the dynamic facility layout problem [J]. 2006, 33: 790 ~ 803.
- [4] Shore RH, Tompkins JA. Flexible facilities design [J]. AIIE Transactions 1980;12(2): 200 ~ 205.
- [5] Gupta T, Seifoddini H. Production data based similarity coefficient for machine-component grouping decisions in the design of cellular manufacturing system [J]. International Journal of Production Research 1990;28:1247 ~ 1269.
- [6] Liggett RS. Automated facilities layout: past. present and future [J]. Automation in Construction 2000;9:197 ~ 215.
- [7] Rosenbaltt MJ. The dynamics of plant layout [J]. Management Science 1986;32:76 ~ 85.
- [8] Lacksonen TA, Ensore EE. Quadratic assignment algorithms for the dynamic layout problem [J]. International Journal of Production Research 1993;31:503 ~ 17.
- [9] Urban TL. A heuristic for the dynamic facility layout problem [J]. IIE Transactions 1993, 25: 57 ~ 63.
- [10] Balakrishnan J, Cheng CH, Conway G. An improved pair-wise exchange heuristic for the dynamic plant layout problem [J]. International Journal of Production Research 2000; 38 (13): 3067 ~ 77.
- [11] Conway DG, Venkataramanan MA. Genetic search and the dynamic facility layout problem [J]. Computers and Operations Research 1994;21:955 ~ 960.
- [12] Balakrishnan J, Cheng CH, Conway DG, Lau CM. A hybrid genetic algorithm for the dynamic plant layout problem [J]. International Journal of Production Economics 2003, 86: 107 ~ 20.
- [13] Kaku BK, Mazzola JB. A tabu search heuristic for the dynamic plant layout problem [J]. INFORMS: Journal on Computing 1997;9:374 ~ 384.
- [14] Baykasoglu A, Gindy NNZ. A simulated annealing algorithm for dynamic layout problem [J]. Computers and Operations Research 2001;28:1403 ~ 1426.
- [15] Erel E, Ghosh JB, Simon JT. New heuristic for the dynamic layout problem [J]. Journal of the Operational Research Society 2003;54:1202 ~ 1275.
- [16] Kochhar, J. S., & Heragu, S. S. (1999). Facility layout design in a changing environment [J]. International Journal of Production Research, 37 (11), 2429 ~ 2446.
- [17] Balakrishnan J, Cheng CH. Dynamic layout algorithms: a state-of-the-art survey [J]. Omega; International Journal of Management Science 1998; 26: 507 ~ 521.
- [18] Balakrishnan, J., Jacobs, F. R., & Venkataramanan, M. A.. Solutions for the constrained dynamic facility layout problem [J]. European Journal of Operational Research, 1992, 57, 280 ~ 286.
- [19] Alan R. McKendall Jr., Jin Shang, Saravanan Kuppussamy. Simulated annealing heuristics for the dynamic facility layout problem [J]. 2006, 33: 2431 ~ 2444.
- [20] Driscoll J, Sawyer JHF. A computer model for investigating the relay layout of batch production areas [J]. International Journal of Production Research 1985;23:783 ~ 794.
- [21] Dorigo M, Maniezzo V, Colomi A. Positive feedback as a search strategy [J]. Technical Report, 91 ~ 016, Politecnico di Milano, 1991.
- [22] Adil Baykasoglu, Turky Dereli, Ibrahim Sabuncu. An ant colony algorithm for solving budget constrained and unconstrained dynamic facility layout problems [J]. The International Journal of Management Science, 2006, 34: 385 ~ 396.
- [23] Kirkpatrick S, Gelatt CD, Vecchi MP. Optimization by simulated annealing [J]. Science 1983;220(4598):671 ~ 680.
- [24] Wilhelm MR, Ward TL. Solving quadratic assignment problems by simulated annealing [J]. IIE Transactions 1987;19(1): 107 ~ 119.
- [25] BaykasogluA, Gindy NNZ. A simulated annealing algorithm for dynamic facility layout problem [J]. Computers & Operations Research 2001; 28(14): 1403 ~ 1426.
- [26] Balakrishnan J, Cheng CH. Genetic search and the dynamic layout problem [J]. Computers & Operations Research 2000, 27 (6):587 ~ 593.

Dynamic Facility Layout Problem Systems Based on Hybrid Ant Colony Optimization Algorithm

LI Hui¹, HUANG Guo-wen², QI Er-shi³

- (1. The college of management and economics, Tianjin University, Tianjin, 300072;
2. The college of management and economics, Tianjin University, Tianjin, 300072;
3. The college of management and economics, Tianjin University, Tianjin, 300072)

Abstract: Facility layout studies mainly address problems related to manufacturing facility layout in order to reduce the material handling costs in the plant. An effective facility layout method can improve the overall operational efficiency during the manufacturing process. With the increasingly fierce competition in the market, manufacturing enterprises must continuously redesign their facility layout in order to adapt to the changing production demand and reduce the cost. The Dynamic facility layout problem (DFLP) could be an effective solution to solve these layout problems.

The total cost of DFLP is comprised of the total material handling cost between facilities and the redesign cost of manufacturing systems during each time. Therefore, the objective function of DFLP is usually defined as the minimization of the sum of the total material handling cost between facilities and the redesign cost of manufacturing system during each time. This paper considers the impact of financial budget on DFLP. In some stages, due to the constraints of financial budget redesigning cannot be actually implemented even if it is beneficial to the manufacturing system.

This paper proposes an improved hybrid ant colony optimization (HACO) model to solve DFLP with budget constraints. Ant colony optimization algorithm (ACO) is a new kind of heuristic algorithms developed in the past few years. In recent years, ACO is used to solve a series of NP-hard problems such as processing plant schedule problem, graph coloring problem, quadratic assignment problem, and vehicle routing problem. The fundamental idea of ACO is based on the behavior of natural ants that succeed in finding the shortest path from their nest to food sources via pheromone trails. The core content of ACO consists of the following three parts: selection, update and coordination mechanisms. Selection mechanism is that the more the pheromone on one path, the more probable the path will be selected. Update mechanism is that the pheromone concentration on one path will increase along with the increasing number of the ants passing through. The pheromone may also evaporate gradually at the same time. Coordination mechanism is that the ants exchange information by means of pheromone and influence each other.

On the basis of ACO, this paper used the simulated annealing (SA) to generate the Hybrid Ant Colony Optimization Algorithm (HACO), and improve the performance of the original ACO. We used 48 problems obtained from Balakrishnan and Cheng to test the efficiency of ACO and HACO proposed. The two algorithms were compared with other typical heuristic algorithms including simulated annealing (SA) proposed by Baykosaglu and Gindy, genetic algorithm (GA) by Balakrishnan, and dynamic programming (DP) by Erel.

SA, GA, DP and ACO algorithms helped locate 2, 12, 13 and 22 optimal solutions respectively based on 48 sets of facility layout. In contrast, HACO obtained 36 optimal solutions. The HACO algorithm proposed by this paper can show good performance for both small and large-scale facility layout problems. There exists a great gap between HACO and the other algorithms in solving large scale facility layout problems. Therefore, HACO is an effective method to solve dynamic facility layout problems with budget constraints.

Key words: dynamic facility layout problem; Hybrid ant colony algorithm; budget constraint

中文编辑: 杜 健; 英文编辑: Charlie C. Chen